Injecting Knowledge into Deep Neural Networks

Sean Quinn, Alessandra Mileo

sean.quinn@insight-centre.org

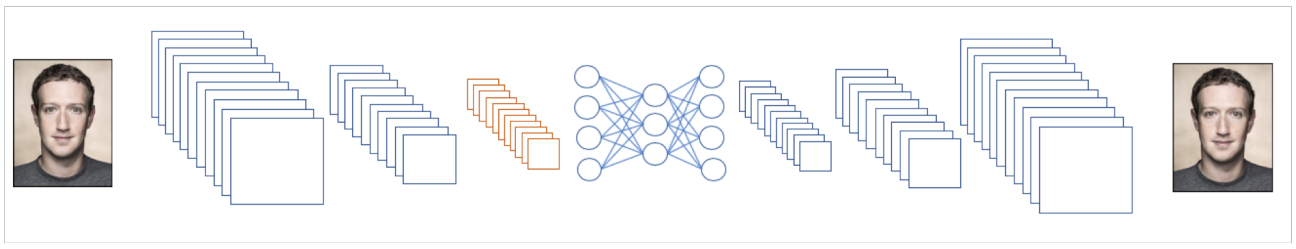
Insight
Centre for Data Analytics

Motivation

Artificial neural networks are currently the most powerful family of algorithms in Artificial Intelligence. The hierarchical structure of a neural network means they are great at extracting high level concepts from low level “raw” data but at present they are not capable of explicitly modelling many of the complex relationships which exist in the real world. Similarly, Neural Networks are somewhat of a “black box”, we know what goes in and what comes out, but we do not understand the decisions which are being made internally in this process. There have been calls within the Artificial Intelligence community to increase the explainability, modelling complexity and reusability of Neural Networks. This research aims to address these issues by injecting and extracting background knowledge into the neural learning process.

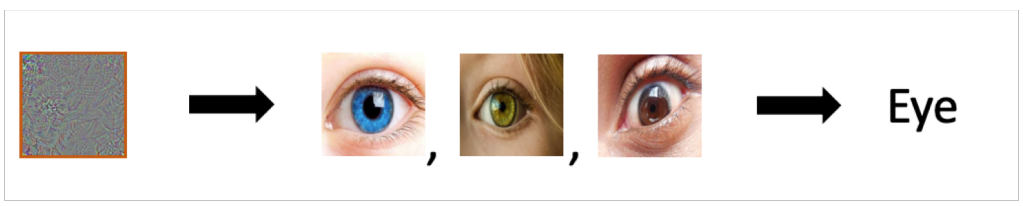
A Sample Use Case: Semi-Supervised Image Recognition with Background Knowledge

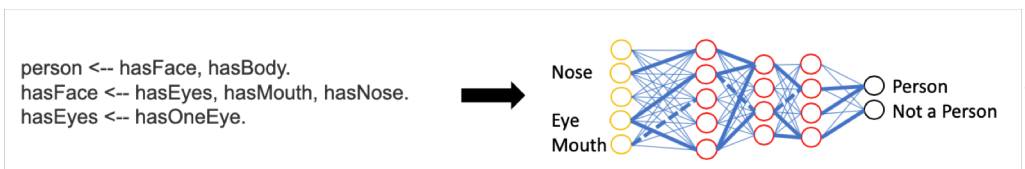
- Train a convolutional autoencoder [1] using both labelled and unlabelled data.



- Select features at a given layer of the network. The ideal candidate layer for selection can be found empirically to ensure the features are semantically meaningful.
- While this example extracts features at a single layer ideally we would aim to incorporate multiple layers to take advantage of the hierarchical structure of the network.

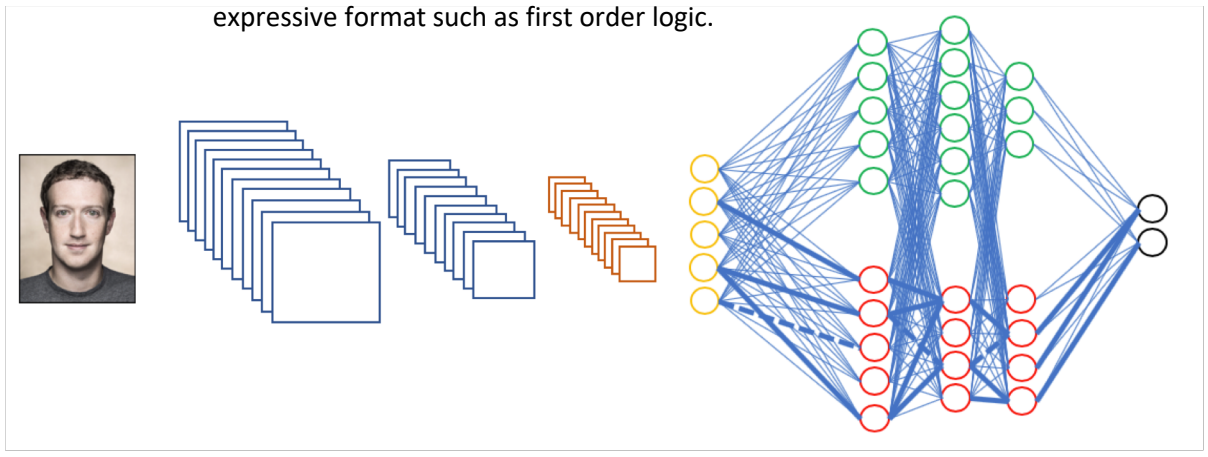
- Re-run data on trained network and for each feature record the image segments which maximise activation. Use these segments to label the feature.
- This example involves manual labelling of the features, future versions will move towards automatic labelling.





- Create propositional rules based on the features which have been identified and the networks outputs and generate a neural structure from the rules [2].
- While we create rules manually in this simple example ideally they would be extracted from a previously trained network or gathered from an ontology and would be in a more expressive format such as first order logic.

- Using the autoencoder as a feature extractor, place the neural format rules in parallel with other fully connected neurons [3] and then train classifier on labelled data.



Impact

By training the network in this way we reduce the amount of labeled data required to perform classification and we explicitly model part of the networks decision making process. On completion of the networks training we can observe the effects of the data on the rules. Have they been weakened / “trained out” or have they been strengthened. Similarly we can observe the creation of new rules. We then aim to extract the refined rule set for future use.

The above example represents first steps towards injecting background knowledge into a modern neural network architecture with many areas of weakness for future development and improvement.

[1] J. Masci, U. Meier, D. Cireşan and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction", *Lecture Notes in Computer Science*, pp. 52-59, 2011.

[2] G. Towell and J. Shavlik, "Knowledge-based artificial neural networks", *Artificial Intelligence*, vol. 70, no. 1-2, pp. 119-165, 1994.

[3] R. Maclin and J. Shavlik, "Creating advice-taking reinforcement learners", *Machine Learning*, vol. 22, no. 1-3, pp. 251-281, 1996.

This project is funded by the Irish Research Council GOIPG/2018/2501

DCU · NUI Galway OÉ Gaillimh · UCC University College Cork, Ireland Coláiste na hOllscoile Corcaigh · UCD Dublin · Science Foundation Ireland sfi For what's next