Paper summary, week 3

David McSharry

1 Deep Residual Learning for Image Recognition

This paper points out that it is surprising that deeper neural networks perform poorer than shallower networks beyond a certain depth, this is unexpected since a neural network could simply be trained to allow certain layers to be the identity. In other words, deeper neural networks must always be able to represent the function of a shallower network. In practice however, this is often not achieved since "learning the identity" is not trivial, weights are taken from a distribution with mean zero and tend to stay this way. This makes learning the identity hard. This paper develops a method of adding the previous layers output to a specific layers output, so that the output of the specific layer (with weights centered around zero) act as a perturbation to the identity. If it would be optimal for that layer to not do anything (in other words act as the identity) then the weights of that layer can simply approach zero as they tend to do, and the output from the previous layer will simply be fed forward, as if acted upon by the identity. This combines the advantages of deeper networks with those of shallower networks.

A non trivial strength of this paper is that it utilises and verifies "the bitter lesson". Smaller depth places a constraint on the neural network and larger depth without residual learning (without applying the identity at each layer) makes good solutions too difficult to find in parameter space. Deep networks with residual learning removes the constraint of the less deep networks while also making it easy for the network to not use them, giving the network more freedom to find the best solution on it's own. (A counter point to this is that making the network MUCH larger makes it perform worse since it becomes feasible to simply learn the training data set, maybe the take away is to generally give neural networks slightly fewer degrees of freedom for it to be able to over fit consistently.)

2 layer normalization

Normalisation is making the mean of a set zero and the biggest and smallest elements of that set 1 and -1. This paper improves on batch normalisation by making it so that the input into neurons in a layer are normalised for each item in a data set. Batch normalisation takes an input and across each item in the batch normalises across a feature that came from the same neuron, this is not suitable for small batch sizes. For layer normalization each individual data point is normalised so that batch size has no effect on this process. Layer normalisation also has an advantage over batch normalisation since it does not depend on batch size, it acts in the same way during training and testing. Because normalisation happens on a per data point basis, layer normalisation is invariant to the shifting and scaling of individual training points. Layer

normalisation speeds up the performance of neural networks.

One non trivial downside to this paper is that there is no normalisation scheme that is invariant under all of the listed transformations outlined in table 1. This means that one must weigh the pros and cons of each scheme as there will be trade offs for not using another, and there is no "one size fits all" solution to normalization.