**Introduction**

The goal of statistical language modelling is to define a probability distribution over a sequence of words (Zhai & Massung, 2016, p. 50) or phonemes/characters. Statistical language models (SLMs) are useful to any natural language application involving prediction, for example automatic speech recognition, machine translation, and text generation. Perhaps the best-known SLM is the *N*-gram, but the largest progress in SLM in recent years can be attributed to recurrent neural networks (RNNs). These are artificial neural networks in which the neurons’ output does not only move forward in the network, but can be looped onto itself or led back to earlier neurons.

RNNs seemed promising for SLM [because …], but were found to be difficult to train, with the parameters often settling “in a sub-optimal solution which takes into account short-term dependencies but not long-term dependencies” (Bengio, 1993, p. 13). A crucial innovation in this respect was long short-term memory (LSTM; Hochreiter and Schmidhuber, 1997): a network architecture in which the neurons…

Since the introduction of LSTM, RNNs have been shown to have great potential as SLMs. Their superiority over other type of SLMs has, among others, been demonstrated by Chelba, et al. (2014). Despite, as a category of models, currently being state-of-the-art, there are still many open questions regarding what the optimal architecture of such models would be. What counts as ‘optimal’ of course is dependent on factors such as the available computational resources, …, and … .

In this report, we will investigate how four parameters relating to RNN architecture affect the quality of the resulting language model. This ‘quality’ is operationalised in two ways: the lowest categorical cross-entropy that the model achieved during training (objective), and the quality of a lyric generated by this model (subjective).

The four parameters under investigation are:

1. Vocabulary size
2. Number of layers (1, 2, 3)
3. Number of nodes (256, 512)
4. Drop-out rate (0.2, 0.5)

In the context of RNNs, the vocabulary size concerns the number of input and output nodes. Seemingly trivial, a decision to both include upper- and lower-case letters implies doubling the number of nodes, making correct classification more difficult.

The number of layers…

The number of nodes…

The drop-out rate…

References

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157–166.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Comput. 9, 8 (November 1997), 1735-1780. DOI=http://dx.doi.org/10.1162/neco.1997.9.8.1