Preston Robertson

IE 8990
Spring 2022
Homework #1
Due Date: 2/8/2022 5PM CST
Submission: Please put your answer and code in a PDF file and upload on Canvas

Q1. Consider the softmax activation function in the output layer of a neural network model, in which real-valued outputs v1, . . . , vk are converted into probabilities as follows:
oi = exp(vi)∑kj=1 exp(vj ) , ∀i ∈{1, . . . , k}

- Show that ∂oi/∂vj = oi(1 −oi) when i = j; ∂oi/∂vj = −oioj when i ≠ j.
  - Quotient Rule:
    - $g_i = e^{v_i}$
    - $g'_i = e^{v_i}$
    - $h_i = \sum e^{v_j}$
    - $h'_i = e^{v_j}$
  - $\frac{e^{v_i}(\sum e^{v_j}) - e^{v_j}(e^{v_i})}{(\sum e^{v_j})^2}$
  - $\frac{e^{v_i}(\sum e^{v_j} - e^{v_j})}{(\sum e^{v_j})^2}$
  - $o_i(1 - o_i)$
- Show that ∂oi/∂vj = −oioj when i ≠ j.
  - $\frac{0 - e^{v_j}(e^{v_i})}{(\sum e^{v_j})^2}$
  - $-o_j(\, o_i)$

- For binary classification problem, assume that we are using the cross-entropy as the loss L = −∑ki=1 yi log(oi), where yi ∈ {0, 1} is the one-hot encoded class label over different values of i ∈{1, . . . , k}. Show that ∂L/∂vi= oi −yi
  -

Q2. Consider a two-input neuron that multiplies its two inputs x1 and x2 to obtain the output o. Let L be the loss function that is computed at o. Suppose that you know that ∂L/∂o = 5, x1 = 2, and x2 = 3. Compute the values of ∂L/∂x1 and ∂L/∂x2.

- $o = x_1 * x_2$
- $\frac{\delta o}{\delta x_1} = x_2$ (vice versa)
- $\frac{\delta L}{\delta x_1} = \left(\frac{\delta L}{\delta o}\right)\left(\frac{\delta o}{\delta x_1}\right) = 5 * 3 = 15$
- $\frac{\delta L}{\delta x_2} = \left(\frac{\delta L}{\delta o}\right)\left(\frac{\delta o}{\delta x_2}\right) = 5 * 2 = 10$

Q3. Consider a neural network with three layers including an input layer. The first (input) layer has four inputs x1, x2, x3, and x4. The second layer has six hidden units corresponding to all pairwise multiplications.

- The output node o simply adds the values in the six hidden units. Let L be the loss at the output node. Suppose that you know that ∂L/∂o = 2, x1 = 1, x2 = 2, x3 = 3, and x4 = 4. Compute ∂L/∂xi for each i.
    - Hidden layer 1 = (x1*x2)
        - Hidden layer 2 = (x1*x3)
    - Sum of HL1 to HL6 = 35
    - o = HL1 + HL2 …
    - $\left(\frac{\delta o}{\delta x_1}\right) = x_2 + x_3 + x_4$
    - $\frac{\delta L}{\delta x_1} = \left(\frac{\delta L}{\delta o}\right)\left(\frac{\delta o}{\delta x_1}\right)$ = (2+3+4)(2) = 18
        - $\frac{\delta L}{\delta x_2} = 16$
        - $\frac{\delta L}{\delta x_3} = 14$
        - $\frac{\delta L}{\delta x_4} = 12$
- How does your answer to the previous question change when the output o is computed as a maximum of its six inputs rather than its sum?
    - HL6 = 12
        - HL6 = (x3)(x4)
    - $\frac{\delta L}{\delta x_1} = 0$
    - $\frac{\delta L}{\delta x_2} = 0$
    - $\frac{\delta L}{\delta x_3} = (4)(2) = 8$
    - $\frac{\delta L}{\delta x_4} = (3)(2) = 6$

Q4. Please select any two models in the Figure 1.11 of the Deep Learning (Goodfellow et al.) book. Discuss why each model was developed, what is the advantage at the time when it was developed, and what is the model limitation.

- Summarize your discussion (1 page, single-space, font size 12pt).
- Please add citations and references if there are any.

Both neural networks chosen were recurrent neural network (RNN) to allow for ease of comparison since both models can/will be tested using the same factors. A RNN is type of neural network that uses supervised learning to perform analysis on time-series data. With the creation of this model, the door opened for deep learning to allow for nonlinear time series data analysis. The first model discussed was some of the motivation behind the creation of the RNN.

Recurrent Neural Network for Speech Recognition is a type of model specifically designed to recognize pattern in speech to allow for computer translation. This type of model had been the goal of many deep learning enthusiast and served as a benchmark for achievement. Though possible in its original form, this model has seen significant improvement as the RNN continues to grow. Why is the RNN the first time the speech recognition dataset had seen good results? This is due to the time-series nature of neural networks allowing for pattern recognition in sound levels. Since the model feeds into itself, it can account for the vibrations from the last five seconds to decide on the sixth. At first this feat was only reasonably possible with the letters of the alphabet and not full words/sentences. This model was first heavily limited due to the significant amount of human expertise needed to design these models and feature engineering. Since each model would need to be trained through manual calculation spanning several hessian matrices for each node, this model was nearly impossible to design. With the improvement of technology, the RNN model no longer needs to be trained manually and can be trained using other methods, making this method much more feasible. The RNN model also saw several improvements over the years, such as the long short-term memory model. This model allows for better retention of long-term data with preventive measures to learning slowdown. The LTSM proved to be a great tool in this endeavor still being used in the best speech recognition models to this day. However, one major limitation to the LTSM is how these very large models were trained using British/American English making it very difficult for different dialects to be understood by the model. Even when re-training the model, different dialects are proving to be a major issue for one RNN model to handle. Other than the LTSM, another alternative model made was the Echo State Network.

The Echo State Network (ESN) was a very early design architecture for the RNN. This model proved to not work well the speech recognition dataset but outperforming in more sporadic datasets such as the stock market. This model was originally created to serve as a design for the RNN to allow for practical supervised learning since the RNN required several man hours to run. The ESN also made advancements into the field of reservoir computing, which make up Liquid State Machines, up to this day. Reservoir learning refers to how a group of nodes are interconnected through a portion of the network. Reservoir topology refers to the layout of said group of nodes. These different topologies include Delay line reservoir (DLR) which is simply a feed forward line of nodes. An alternative to DLR is DLR with feedback connections, this model is very similar to the previous model, but each node is connected to the nodes preceding it. The final topology I will discuss is the simple cycle reservoir, which is a group of nodes connected in a cycle pattern. Each of these topologies can be mixed into the same ESN since each ESN can have multiple sub-reservoirs. The limitations of the ESN include that it is a discrete time based RNN which means it can not learn data live/continuously. Fitting an ESN can be very time consuming and require other methods to tune each reservoir. This model runs into an overfitting issue that is present in all RNN models but especially with ESN models. To counteract these limitations, the community have come up with several modifications to the original ESN since its conception. The Growing ESN is a complicated process that hopes to help train the model through matrix operations and linear algebraic properties to adjust the weights of each node. The next improvement hopes to help improve training time by changing the original ESN models Bayesian regression approach to a Laplace likelihood function. This Robust ESN would allow for faster training times cutting out the cross-validation step.

Overall, both models have proven extremely useful in the process of learning and growing RNN, both in their original models needed a lot of improvement to make it to where they are now. With that in mind, the LTSM and normal RNN seem to have more traction within the community and seem to work better than the ESN. The LTSM is just more equipped to handle complicated problems and with the limitations

of the ESN and its non-standard training, has the model with a lot to be desired. The ESN does not see much use but helped kickstart the reservoir training section of deep learning, so its impact is still substantial.

References:

Ahmad, A. M., Ismail, S., & Samaon, D. F. (2004, October). Recurrent neural network with backpropagation through time for speech recognition. In IEEE International Symposium on Communications and Information Technology, 2004. ISCIT 2004. (Vol. 1, pp. 98-102). IEEE.

A. Rodan and P. Tino, "Minimum Complexity Echo State Network," in IEEE Transactions on Neural Networks, vol. 22, no. 1, pp. 131-144, Jan. 2011, doi: 10.1109/TNN.2010.2089641.

Graves, A., & Jaitly, N. (2014, June). Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning* (pp. 1764-1772). PMLR.

Li, D., Han, M., & Wang, J. (2012). Chaotic time series prediction based on a novel robust echo state network. IEEE Transactions on Neural Networks and Learning Systems, 23(5), 787-799.

Qiao, J., Li, F., Han, H., & Li, W. (2016). Growing echo-state network with multiple subreservoirs. IEEE transactions on neural networks and learning systems, 28(2), 391-404.