STAT 6685 HWK6

Cody Grogan
A02313514

December 9, 2023

Problem 1

1)

Both of these statements are true because the arbitrary accuracy comes with sufficiently large amount of data. Thus if there was sufficient data the network could learn the features itself. However, data is usually finite. Feature engineering can be used when data is finite to push the network in the right direction by doing some of the computations beforehand.

2)

Self attention works by gathering the information about the input before acting on it. This is accomplished through querys, keys, and values (W_Q, W_K, W_V) matrices. All of which are $\in R^{e \times d_k}$. Where e is embedding length and d_k is some arbitrary value. All of these matrices are multiplied by the input matrix $X \in R^{w \times e}$ to yield Q, K, and V ($\in R^{w \times d_k}$) matrices which hold the information about the input.

The attention operation is then done with the Q, K, and V matrices,

$$Z = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Where Z is the hidden representation of the input sequence. Looking just at the first operation,

$$\frac{QK^T}{\sqrt{d_k}} \tag{2}$$

This yields a w by w matrix where each entry in row i and column j shows the strength of the relation between embedding i and embedding j. Then soft maxing along the rows shows this importance between 0 and 1. We then multiply by the value matrix which holds a mapping of the words and the softmax acts as a mask for which words are actually related.

1 Problem 2

Description 1

- Loop through all of the intentions and load the data
- For each entry in the data collect text for the utterance
- add the utterance to the data frame with the intention as the label

Description 2

- We convert the labels to indices so we can use a 1 hot vector for the classification

Description 3

a)

- A tokenizer is a program that splits a sentence and words into a number encoding that preserves the information in a given sentence

b)

- The tokenizer takes base letters and tracks their frequency in the text.
- The tokenizer then makes tokens for combinations of the base letters based on frequency until a maximum vocabulary is reached

Description 4

- The Roberta module is being initialized

Description 5

- CLS indicates to Roberta that goal of the input is to classify it
- SEP token indicates to Roberta the cut where one sentence is no longer related to the next

Description 6

- We use cross entropy loss because we are trying to classify the outputs
- We dont use a softmax because cross entropy loss already has a softmax in it

Description 7

- Doing the typical epoch loop
- Sending the data to the gpu for execution
- Forward pass and get the prediction
- Get the predicted class for each output using max along dimension 1 because softmax would be highest anyway
- Compute loss and do optimizer step
- For the test dataset check how many correct predictions the network has made using max()

Description 8

- This function essentially converts an input sentence into tokens
- Runs input sequence through network
- Returns the label from the network

Given Sentence report

Input	Label
play radiohead song	Play Music
it is rainy in Sao Paulo	Get Weather
sun shinnes all day	Get Weather
low humidity, high altitude	Search Creative Work
Book tacos for me tonight	Book Restaurant
Book a table for me tonight	Book Restaurant
I want BBQ tonight	Play Music

Table 1: Replies for given sentences

New sentence report

Who wrote the book the Martian?	Search Creative Work
Book me a table at Song Bird	Book Restaurant
What is the weather in Boston	Get Weather
Add shine on you crazy diamond to my playlist	Add To Playlist
Is where the red fern grows a good book?	Search Creative Work

Table 2: Responses for made up questions

Chat bot

An intent classifier could be used in a chatbot as an easy way to determine a the tone of the person chatting. This can then be used to change the sentiment of a response from the chat bot. For example the if a customer is angry the response could be modified to seem more apologetic and such.

Problem 3

3.1

Description 1

- What the sliding window function is doing is storing sequence length entries of the data as inputs
- It is then setting sequence length+1 as the target for the input.
- Effectively giving a window of input values for the LSTM to predict the single next value.

Description 2

- We need h_0 and c_0 because we need to initialize the hidden state and context for the lstm as they will be propagated through the gates of the LSTM

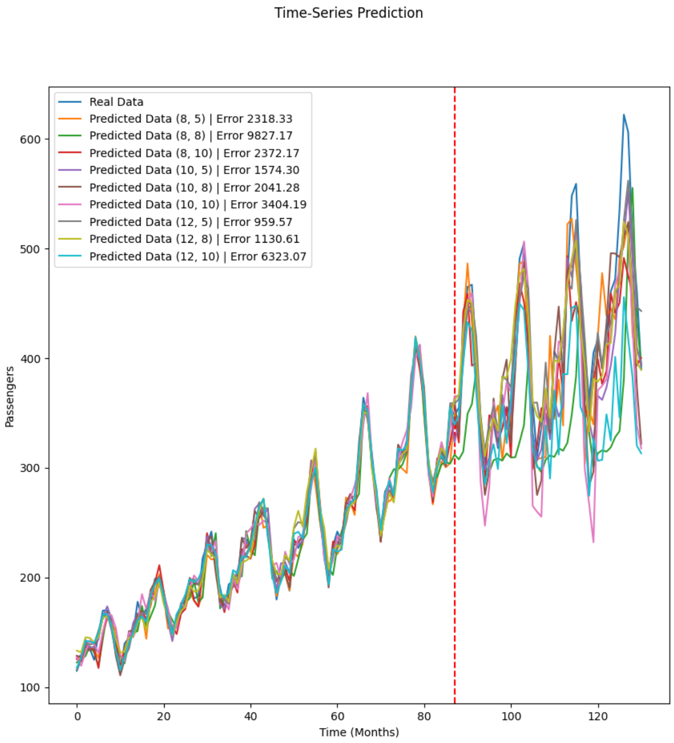
Description 3

- The input size is equal to 1 because the LSTM is executed in a loop for each entry in the input sequence
- Meaning the first value of the input sequence will be executed and the hidden and context will be used on the next value of the input sequence until the end is reached.
- Number of classes is 1 because the we want to make a single prediction from the FC layers

Description 4

The prediction on the test data has significant spikes much below the real value. This is likely because the LSTM overfit to the training data causing it to not generalize well.

3)



Looking at the data where the legend is (sequence length, hidden size) and the loss untransformed before evaluation, we can see that in general increasing the input window increases the overall accuracy of the LSTM. However, increasing the hidden size makes the LSTM overfit in less time.

Problem 3.2

Description 1

- The class first loads the MNIST dataset and normalizes it
- The class then randomly chooses 2,000 images from the dataset and adds a gaussian blur and rotation to each image in domain 2.
- Then if a reshape is requested the transformed images are resized to the requested shape
- The class then stores the regular images as domain 1 and the blurred and rotated images as domain 2

- This operation must be done because the 2 domain representations must be made for the discriminator to learn the pictures within the dataset and the generated ones.

Description 2

- A generator is a function that takes an input and tries to create an image that would be in the dataset.
- A discriminator is a network that tries to determine if a given input is within the dataset or not.
- The output of the discriminator is 1 because its output is true or false

Description 3

- D1 and D2 are the discriminators which determines if a unrotated image is in the dataset and the rotated image is in the rotated dataset
- G1 and G2 are generators in transforming a domain 1 image into domain 2 and vice versa.
- The forward function uses the generators to bring an input image from $1 \longrightarrow 2 \longrightarrow 1$ and vice versa.
- The function also returns a correspondence picture where images are only taken to the other domain.

Training Description

Discriminator

- The discriminator training function generates images from the generators
- The generated images and the original images are then passed to the discriminators
- The discriminators are then updated based on the correctness of their judgements.

Generator

- The generator training function allows the generators to generate images from the test images.
- The generated images are then passed to the discriminators
- The generators are then give a loss based off of reconstruction accuracy and whether they fooled the discriminator

- The correspondence loss is also calculated where a s1 and s2 are the same images but in domain 1 and 2. The loss is then calculated by comparing s2 and the result of a transformation of s1 to domain 2. And vice versa
- The losses are then all added together with a regularization coefficient for the correspondence loss.

Problem 3.3

2)

The main difference between the convolution and attention layers is that the attention layers allow messages to be passed from any node to another node. This is different from the convolution because only messages from neighbors can be passed in a graph convolution.

3)

The accuracy of the GCN was nearly double that of the MLP at 0.8. The tnse plot shows that the GCN is separating the points fairly well but it seems to be unable to classify the teal points affectively.

4)

The accuracy of the GAT network was slightly better than the GCN at 0.817. It also seems that visually there is better separation of the classes overall in the tnse plot with the GAT layers.