Polynomial Expansion Report

Ankit Garg
ankitgarg2123@gmail.com

Data Analysis

- 1 M training data points were provided. This was divided into 60%, 20% and 20% for training, validation and testing

Model Architecture

- I model this problem as sequence to sequence problem
- I used character embeddings which are randomly initialized and are added to positional embeddings and then fed as input to the transformer
- I use exact transformer encoder decoder model as proposed in the [Attention Is All You Need (arxiv.org)](#) paper
- The motivation behind using this architecture is because of its attention mechanism which I believe is essential for a model which can predict the expansion of the factors

Regularization

- I wanted to use early stopping as the regularizer but in the 20 epochs of training I did not observe the model overfitting training data. Hence **no regularization** was required

Prediction Accuracy

Dataset	Accuracy
Training	99.28%
Validation	99.13%
Testing	99.14%

- Very high accuracy was observed on all training, validation and test sets

Parameters

- Total number of trainable parameters are ≈ 5.5 Million

Hyperparameters

Hyperparameter	Value
Embedding dimension	256
Num of attention heads	4
Num of encoder layers	3
Num of decoder layers	3
Feedforward dimension	1024
Batch size	64

Training

```
# SGD optimizer
optm = optim.SGD(neural_net.parameters(), lr = 0.1, momentum=0.9)
# Exponential Decay Scheduler
scheduler1 = torch.optim.lr_scheduler.ExponentialLR(optm, gamma=0.9)
Epochs = 20
```

Conclusion

As we can observe, the model achieves very high accuracy on all training, validation and test datasets. This is expected as the transformer is a very powerful model.