Enhancing NanoGPT via Squentropy Loss Function and Hyperparameter Tuning

Sujay Talanki `stalanki@ucsd.edu` Sujen Kancherla `skancherla@ucsd.edu`
Rehan Ali `rmali@ucsd.edu` Akshat Muir `akmuir@ucsd.edu`
Mentor: Mikhail Belkin `mbelkin@ucsd.edu` Mentor: Yian Ma `yianma@ucsd.edu`

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

The Big Question

Can optimizing the squentropy loss function and adjusting pertinent hyperparameters improve baseline NanoGPT performance?

Mathematical Basics

Squentropy loss is a hybrid loss function that combines aspects of cross entropy and mean squared error.

Consider the following notation:

Let $D = \{(x_1, y_1), ..., (x_n, y_n)\}$ denote the dataset sampled from a joint distribution $D(X, Y)$.
For each sample $i, x_i \in X$ is the input and $y_i \in Y = \{1, 2, ..., C\}$ is the true class label. The one-hot encoding label used for training is $e_{y_i} = [0, ..., 1, ..., 0] \in \mathbb{R}^C$.
Let $f(x_i) \in \mathbb{R}$ denote the logits (output of last linear layer) of a neural network of input x_i, with components $f_j(x_i)$, $j = \{1, 2, ..., C\}$.
Let $p_{i,j} = \frac{e^{f_j(x_i)}}{\sum_{j=1}^{C} f_j(x_i)^2}$ denote the predicted probability of x_i to be in class j.

Then the squentropy loss function on a single sample x_i is defined as follows:

$$L_{squen}(x_i, y_i) = -\log p_{i,y_i}(x_i) + \frac{1}{C-1} \sum_{j=1, j \neq y_i}^{C} f_j(x_i)^2$$

The squared loss portion of L_{squen} acts as a *regularization* term.

Hyperparameter Tuning

Andrej Kaparthy (creator of the NanoGPT repo) states that the current set of hyperparameters utilized are have not been tuned for optimal performance! We have decided to focus on the learning rate, dropout percentage, and number of layers in the neural network. Here are their potential values:

- Lr - 0.00006
- Number of Layers - 16
- dropout - 0.1

Cross Entropy Baseline

The standard loss function used in NLP token prediction scenarios is cross-entropy. Our GPT-2 model trained on Tiny Stories performed well on simple cross-entropy without any hyper-paramter tuning. The model converged with around 1.8 loss value starting at around 10. The perplexity of the model is around 3.8 for the baseline using simple crosss-entropy.

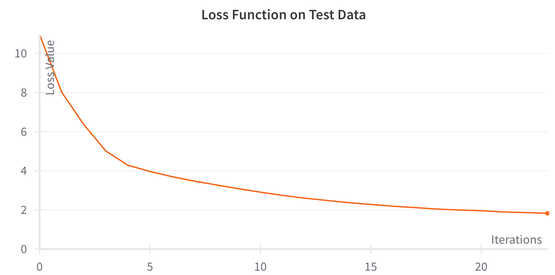

Figure 1: Performance improvement graph showing model convergence with cross entropy loss.

The Great Connection

Both Squentropy and Cross Entropy are able to converge during training process and decrease the loss over iterations significantly well as shown in the Figures. But converging and decreasing loss does not always means increased performance.

Example Story (Cross Entropy)

Once upon a time, there was a little girl named Lily. She was so tired because she didn't want to play. But before she started to feel dizzy, she started to feel a little better.

Suddenly, she heard a noise outside. She looked up and saw a little mouse named Tom. He said, "Hi Lily, what are you doing?"

Lily replied, "I'm just playing!"

Tom looked up and said, "I'm teasing you. Can I try again?"

Lily was so happy to hear this and said, "Sure, you can try another game next time." Tom was so happy! He smiled and said, "Thanks for letting me play."

Lily and Tom continued to play together until the sun started to go down. Then they went back to their homes and Lily cried. "Thank you for the game, Tom!" The end.

Squentropy Best Performance

After tuning the hyperparameters with the squentropy loss, we were able to get a model to converge at around 2.0 from starting at 11 in the loss value. The performance on the test during training is below.

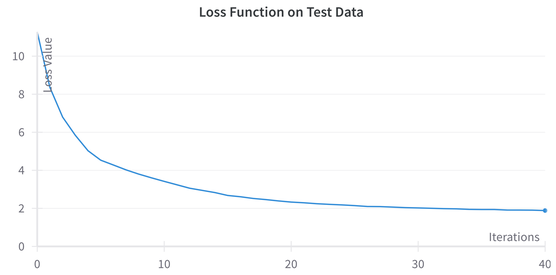

Figure 2: Performance improvement graph showing model convergence with squentropy loss.

Example Story (Squentropy)

Once upon a time, there was a big dog named Max. Max was very grumpy because he did not like to play with his friends. One day, Max's friends wanted to play a game of catch with Max's friends. They all ran to the pitch and started to play.

Max was very good at catch the ball very far. He didn't know that his friends would try to catch it and get it back. His friends were happy to hear him and wanted to play too. Max felt happy that his friends were happy too.

As they played, Max accidentally hit a big tree. His friends tried to help him but Max was still grumpy. Finally, his friends got cross and Max was very happy. His friends were proud of him for being good and telling the truth. Max learned that being grumpy is not a good thing to do. From then on, Max and his friends played with other dogs and had a lot of fun together. The end.

Conclusions and Outlook

Since the best perplexity out of all of the tuned Squentropy models was 5.2 this indicates that although our new model isn't too far off from 3.8 (the results of the Cross-entropy trained model), it does not supersede let alone match its effectiveness. In addition, the text output of the Squentropy model was a lot less intelligible than that of the Cross-entropy model. The future implications of our findings, show that other regression-based loss functions combined with cross-entropy that could yield smaller training loss per iteration whilst matching/beating the perplexity of only cross-entropy trained Large Language Models(LLM).

For Further Information

For the details of our work:

AIx, Y., *Secrets in Training a Large Language Model*, Available at: `https://medium.com/@YanAIx/secrets-in-training-a-large-language-model-bbb0f2472e2f` (2024).
Karpathy, A., *nanoGPT: Minimal GPT-like training code*, GitHub repository, Available at: `github.com/karpathy/nanoGPT` (2024).

References

[1] Hui, Mikhail, Belkin, M., & Wright, S., *Cut your Losses with Squentropy*, arXiv:2302.03952 [cs.LG] (2023). Available at: https://doi.org/10.48550/arXiv.2302.03952.

[2] Hui, Mikhail, & Belkin, M., *Evaluation of Neural Architectures Trained with Square Loss vs Cross-Entropy in Classification Tasks*, arXiv:2006.07322 [cs.LG] (2021). An extended version published at ICLR2021 with added evaluations of Transformer architectures. Available at: https://arxiv.org/abs/2006.07322.

Acknowledgements

This project was created for our pentultimate DSC Capstone Project. Our topic was chosen by our mentors. Website: