DD2424 - Assignment 4 (Bonus)

Summary

I have completed three of the four suggested bonus extensions, using `Adam` instead of `Adagrad`, splitting the into chunks and randomizing them during training, and finally implementing temperature and nucleus sampling to attempt to generate more realistic text.

The code for the assignment has been written in `python`. I have implemented the neural network as a class. For the hand-in, all of the code has been put toghether in a main file with all the functions and the class declared at the top. For the hand-in, I have also commented out the saving of generated figures and results in JSON files, as well as omitting some of the case-specific testing and gradient testing.

Oskar Stigland

DD2424
Spring 2023

`Adagrad` versus `Adam`

I first ventured into testing the effect of replacing the `Adagrad` optimizer with the much-popular `Adam`, for which we compute the updates as

$$\begin{aligned}
\boldsymbol{w}_{k+1} &= \boldsymbol{w}_k - \eta \frac{\tilde{\boldsymbol{m}}_k}{\tilde{\boldsymbol{v}}_k + \epsilon} \\
\tilde{\boldsymbol{m}}_k &= \boldsymbol{m}_k \times (1 - \beta_1^t + \epsilon)^{-1} \\
\tilde{\boldsymbol{v}}_k &= \boldsymbol{v}_k \times (1 - \beta_2^t + \epsilon)^{-1} \\
\boldsymbol{m}_k &= \beta_1^t \times \boldsymbol{m}_{k-1} + (1 - \beta_1^t) \times \nabla_{\boldsymbol{w}} L \\
\boldsymbol{v}_k &= \beta_2^t \times \boldsymbol{v}_{k-1} + (1 - \beta_2^t) \times (\nabla_{\boldsymbol{w}} L)^2
\end{aligned}$$

where ϵ is a very small numnber and t is usually updated per training batch. Further, we have that $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In this case, I have used $\eta = 0.01$ for both `Adagrad` and `Adam`, and since the data will be split into randomized chunks, t is updated after each chunk, usually corresponding to some 500 training examples. In order to properly check the effect of changing optimizer, I have also carved out the last 5% of the training examples for validation, and a smooth validation loss is tracked along with the smooth training loss. For the former, a random sample is selected after each training example and both smooth loss are then updated according to:

$$\tilde{\ell}_t^{\text{train}} = 0.999 \times \tilde{\ell}_{t-1}^{\text{train}} + 0.001 \times \ell_t^{\text{train}}, \quad \text{and} \quad \tilde{\ell}_t^{\text{val}} = 0.999 \times \tilde{\ell}_{t-1}^{\text{val}} + 0.001 \times \ell_t^{\text{val}}$$

The difference in training performance when using `Adagrad` and `Adam` is shown in the figure below. There is a clear distinction between the two optimizers, and the main point seems to

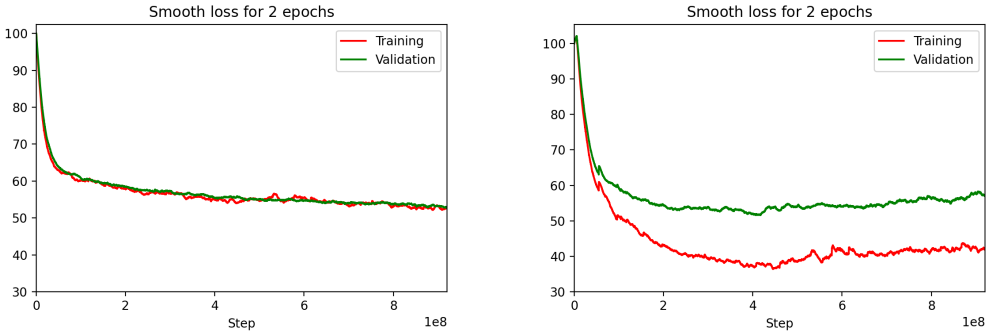

Figure 1: Training and validation loss for `Adagrad` (left) and `Adam` (right), after 2 epochs of training and $\eta = 0.01$.

be that in this particular example, `Adam` is more prone to overfitting. The training loss is much lower in the latter case, but the validation loss is very similar. That is, both models seem to be equally good at generalizing to unseen data, while `Adam` achieves a much lower loss on the training dataset. In fact, the validation loss for `Adam` seems to increase somewhat towards the end. Hence, in this limited example, `Adam` does not seem to offer any actual benefit over `Adagrad`.

Training with randomly ordered chunks

In the first experiment, the data was iterated through sequentially. However, this might introduce temporal dependencies, for example skewing the validation loss whenever we reach the end of an epoch (since the validation loss is extracted as the last $\sim 5\%$ of sequences. Instead, it seems more sensible to train on randomly selected sequences. This does, however, introduce the problem of memory and context and would require the h_0 to be reset at every training step. Thus, I split the data into 100 blocks, save the last 5 for validation and shuffle the rest at the beginning of each epoch, such that h_0 does not have to be reset at each training step - but instead only at the beginning of each block of training examples. The results are shown for both `Adagrad` and `Adam` in the figure below. The results are more or less identical to those obtained without

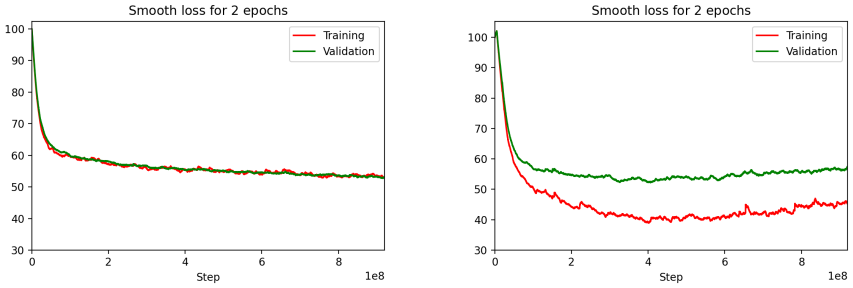

Figure 2: Training and validation loss for `Adagrad` (left) and `Adam` (right), after 2 epochs of training and $\eta = 0.01$ with randomly ordered blocks of training sequences.

randomly ordered blocks of training sequences. That is, randomizing the sequences does not seem to improve the ability of the model to learn or to generalize significantly much better to new data. In order to possibly improve this, I train another set of models with $T = 50$, $m = 200$, and using `He` initialization. Further, for the last model with `Adam`, t is incremented for every second block (rather than at every block). All loss metrics have been normalized with respect to the sequence length to allow for a proper comparison. The results do seem to show that is it possible to achieve a lower loss with the combination of randomized blocks sequences, a wider net, `He` initialization and a longer sequences, e.g. $T = 50$.

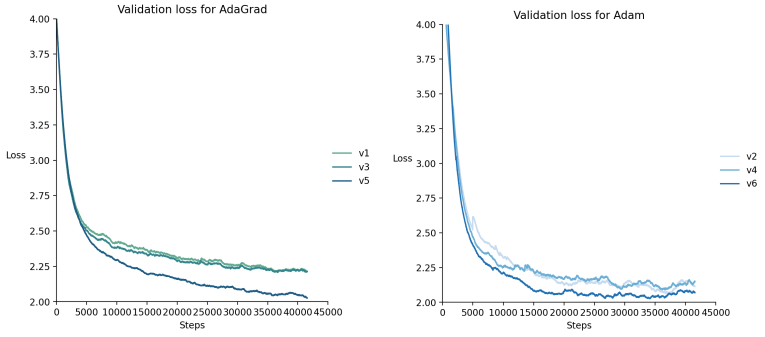

Figure 3: Validation loss for models with `Adagrad` (left) and `Adam` (right). v1 and v2 are initial models with ordered training sequences, v3 and v4 are trained with randomized blocks with $T = 25$ and $m = 100$. v5 and v6 are trained with $T = 50$, $m = 200$ and with `He` initialization.

Generating (more) realistic text sequences

First, I drew upon the results from the two previous experiments to train a better model. Due to the possible overfitting problems with `Adam`, I chose to pursue a model similar to v5, using $T = 50$, $m = 200$, `He` initialization and a longer training period: 10 epochs with an `AdaGrad` optimizer and $\eta = 0.01$. The training results are displayed in the figure below. In what follows,

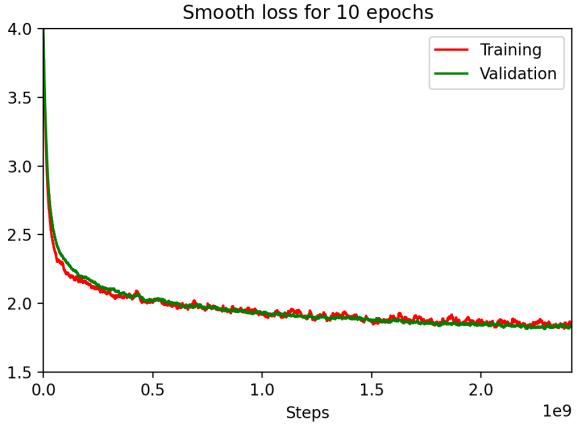

Figure 4: Training and validation loss for `Adagrad`, after 10 epochs of training with $\eta = 0.01$, with $T = 50$ and $m = 200$ with `He` initialization.

I will display some examples of generated text when using varying temperature values, τ, and varying thresholds, θ. Since the effects τ and θ are in a sense proportional - i.e. a high θ has an effect similar to that of a hihg τ, and vice versa - I have attempted to separate the effects.

Varying θ

$\tau = 1.0$, $\theta = 1.0$

> Prgvenscr Potter and to the gtelled the store had sthe ded the sare of the sare goo the squited the start of the tto- he had sof Harry and thing the stareved thee sard. "I know and in be to the soulang the swat had she roblanging ton the store had Moode the starm pounend the stare 's the stared andi

$\tau = 1.0$, $\theta = 0.9$

> Perthe dol har sere spreton's get, theis hen and tonked eye an the milly - y orared ant her thent bonden me eroce a down courny, and Harry her gails fice foo headed for to ste sivee, you welle way send orfurise, a the seld gooren moven's got dowe inth anothers. He avery fee sere.
> "No Hormaris. "Ho

$\tau = 1.0$, $\theta = 0.7$

> Prtthe don the said to her the tome to do the pouster the camped to his hand the stoply he said and her was a bating and he was stinked to seat the cams of the

compinge stad it and stared and Harry was here sood bof the stare the rousher sower stinger, the ground the cange the could be she and the st

$\tau = 1.0$, $\theta = 0.5$

Prtthe don the said to her the tome to do the pouster the camped to his hand the stoply he said and her was a bating and he was stinked to seat the cams of the compinge stad it and stared and Harry was here sood bof the stare the rousher sower stinger, the ground the cange the could be she and the st

Varying τ

$\tau = 0.9$, $\theta = 0.9$

Prgcaniden. "The doring about to and pistly an the Gobriolly looked all.
" Ren lonked looked fithor coured you, Hermione thome ther hinge, had tering and way, and Ron facting firming and bagis, with a arling to soupre sach. Harry capple corit to be hadded has peatering stupe to the wit his mond hi

$\tau = 0.7$, $\theta = 0.9$

Prrto gaven sumported a thank in ort he to good the nopred the heard for and mearly that could wet stering beand seave sterngs and he there of the fains as the Dark adound in of Mosint, Harry had with as she the bound stere as the said have le to the off corss at a sire beat and to the racked the was

$\tau = 0.5$, $\theta = 0.9$

Prof sadny, shound a lang that he said the grint of the wiss the sered the call and the was staking the seeped and har been to the cared and the beared somenging as the coully a berting the bating to kis the pond and the grint his panted and the rect a hinge the fored to the tor the dade the stiding

Conclusions

Varying θ and τ clearly have similar effects. As we decrease the parameter values, the generated becomes more "deterministic", and the variation in text decreases rapidly. The model generates fewer special characters and line breaks and the text seems overall to be less "expressive". It seems an "optimal" parameter set is something like $\theta = 0.9$, $\tau = 0.7$, in order to generate fairly expressive text containing some keywords and names, such as "Ron", "Hermione", "Harry", and "Potter". However, the larger problem is that the model has a difficult time generating actual words using character sequences. It seems difficult to get more than marginal improvements from more hidden units and longer sequences when employing an RNN. Rather, it might be necessary to extend the model to e.g. an LSTM or a GRU, which is a kind of "simplified" LSTM. It might also be helpful to employ embeddings instead of the sparse categorical inputs attained from one-hot encoding vectors.