Meeting — Summary LLM

Bas Donkers, Dennis Fok, Finn-Ole Höner

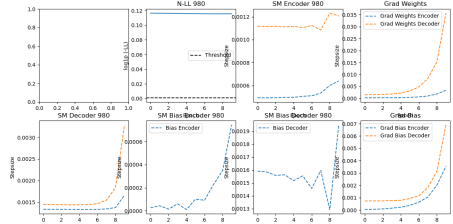
May 22, 2024

Updates

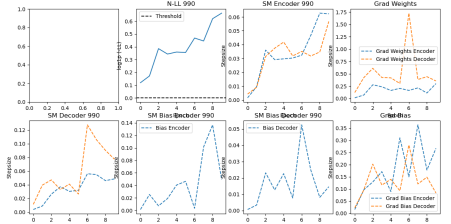
- Gradient spikes first, then large step in decoder / encoder
- The other gradients seem to react to this jump
- Gradient $\rightarrow$ Step $\rightarrow$ Other gradients $\rightarrow$ Step $\rightarrow$ -LL jumps
- `torch.nn.utils.clip_grad_norm_(ae.parameters(), max_norm=dMaxGrad)`[1] seems to help. This clips the gradient if the norm exceeds `dMaxGrad`. See here[2].

[1] https://pytorch.org/docs/stable/generated/torch.nn.utils.clip_grad_norm_.html#torch.nn.utils.clip_grad_norm_

[2] https://www.cs.toronto.edu/~rgrosse/courses/csc321_2017/readings/L15%20Exploding%20and%20Vanishing%20Gradients.pdf#page=6.78



(a) Epoch 970-980



(a) Epoch 980-990

Figure 2: Spikes: Notice how the decoder gradient norm increases first, causing a large step and in-turn affecting the other gradients. The scales of the y-axis differ across all plots.

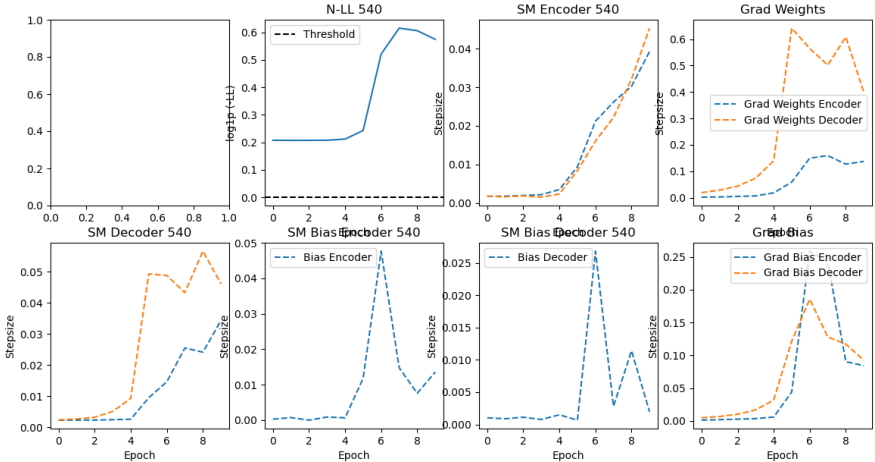


Figure 3: More spikes I

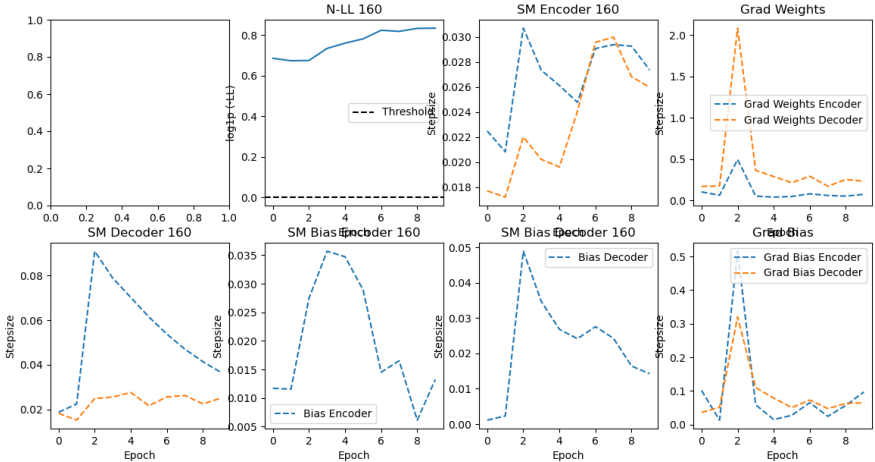


Figure 4: More spikes II

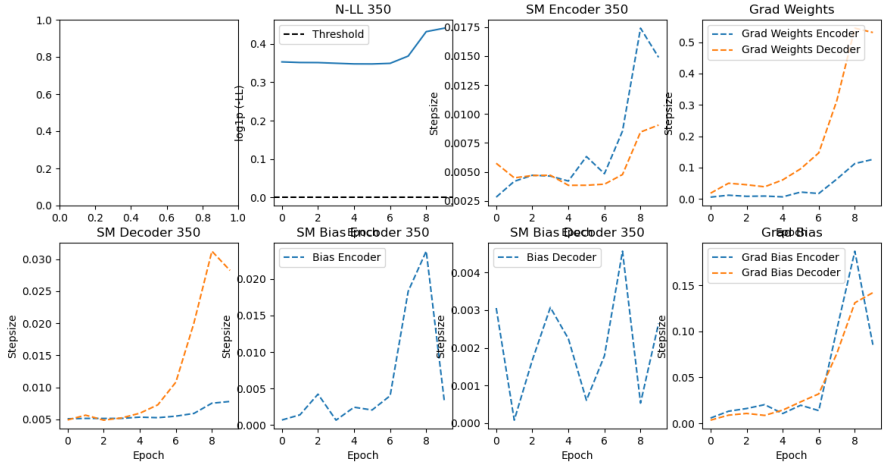
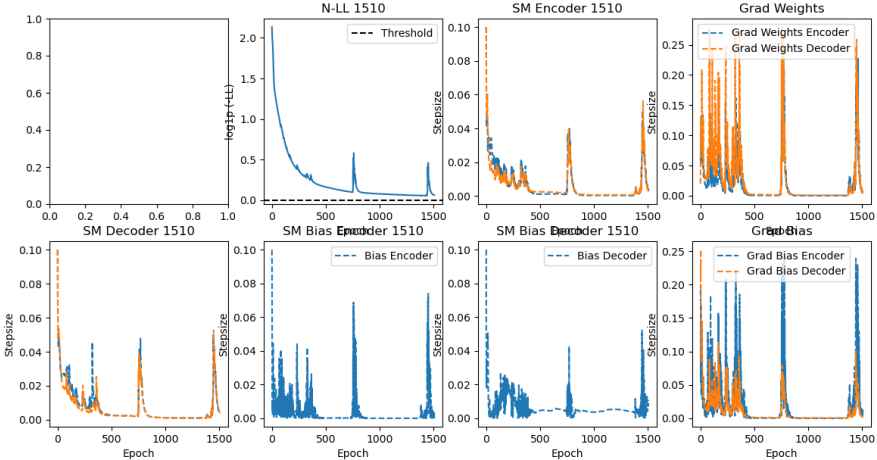
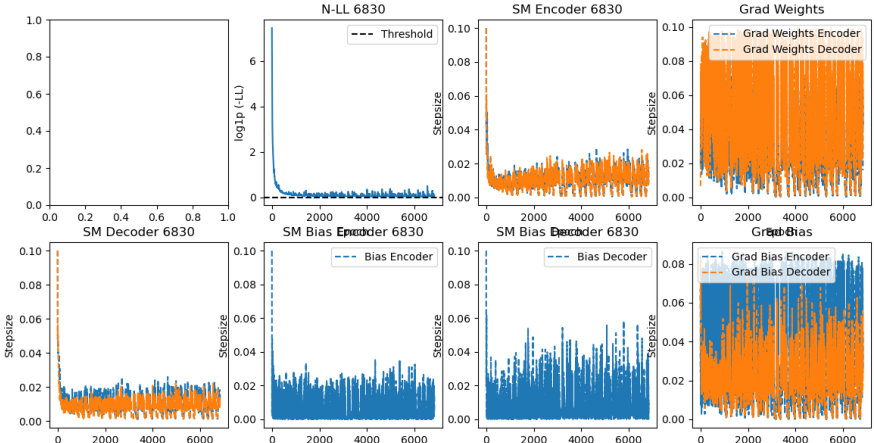


Figure 5: More spikes III

Optimization with gradient-clipping at $dMaxGrad = 0.3$.



Optimization with gradient-clipping at $dMaxGrad = 0.1$. Deeper in the optimization those spikes seem to become a problem again, at a lower scale. Oscill. of gradients later in the optimization.



Optimization with gradient-clipping at $dMaxGrad = 0.05$. This model also has a normalization layer after the hidden layer.