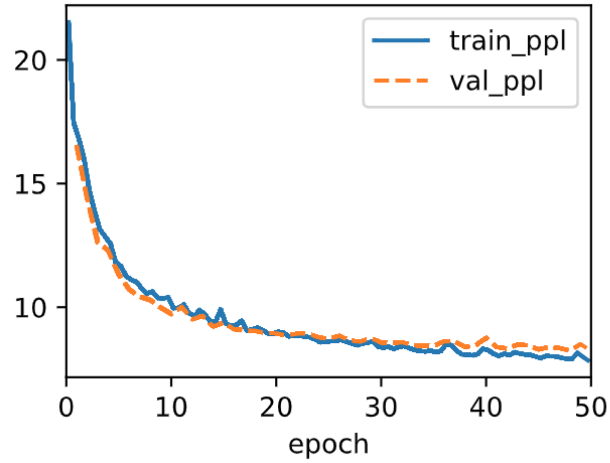
Jacinto Martinez | Student ID: 801135016

HW4

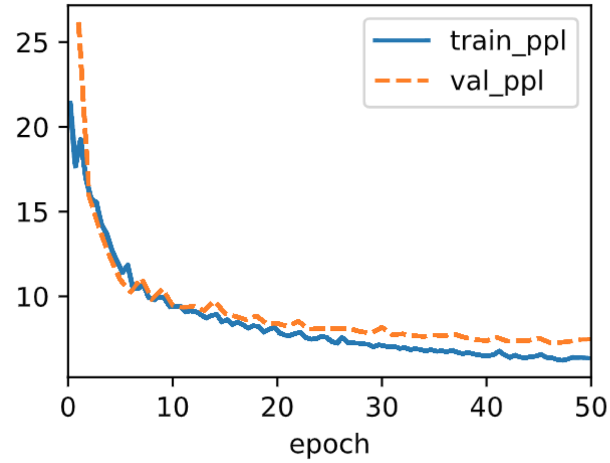
Github Link: https://github.com/jacintomart/4106/tree/main/HW4

Problem 1a

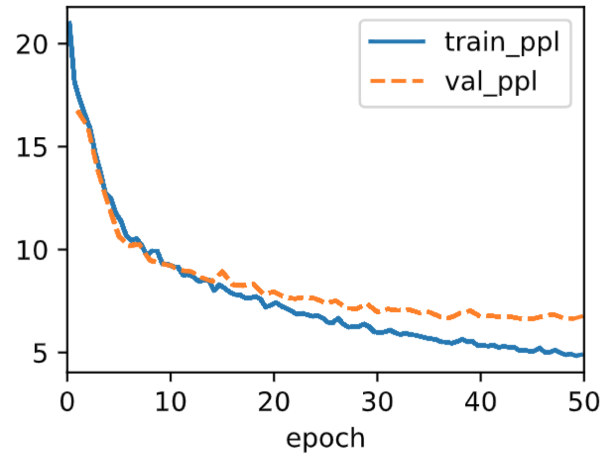
GRU-8 State



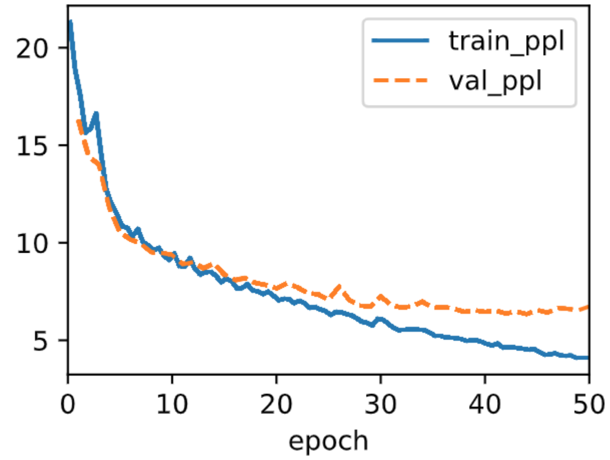
GRU-16 State



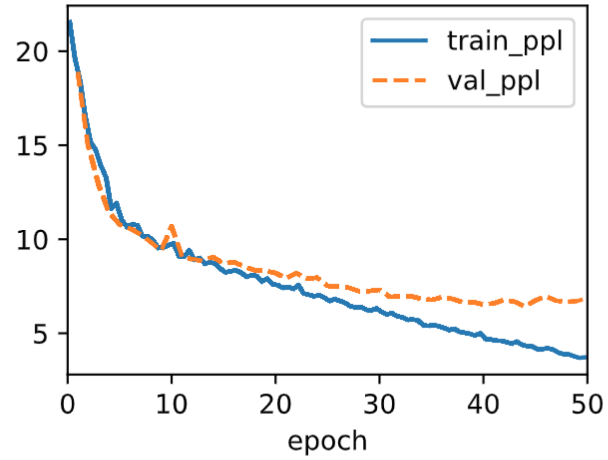
GRU-32 state



GRU-64 state



GRU-128 state



As the number of hidden states was increased across the GRU models, the overall performance experienced a steady improvement with decreasing perplexity. However, this improvement in performance came at a cost, with model complexity and size both roughly tripling with every doubling of hidden states. The table below compares the performance metrics of each of the 5 models trained, with a lower perplexity rank indicating lower perplexity.

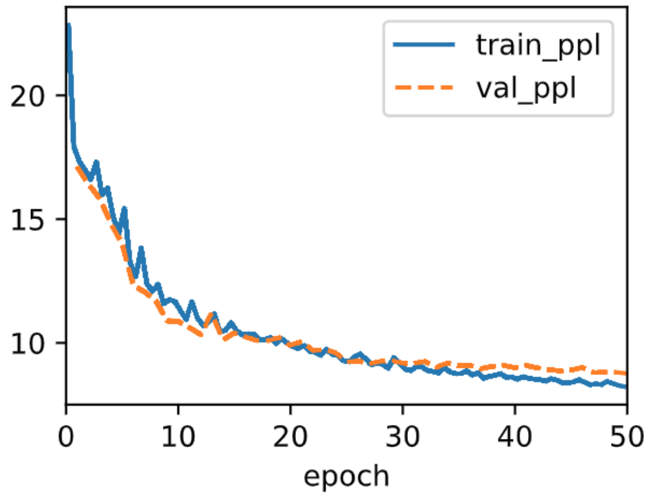
Model	Perplexity Rank	Train Time	Size (Params)	Complexity (MACs)
GRU-8	5	1min 29s	912	991.23k
GRU-16	4	1min 31s	2.21k	2.38M
GRU-32	3	1min 44s	5.95k	6.32M
GRU-64	2	2min 43s	18.05k	18.94M
GRU-128	1	5min 33s	60.67k	63.05M

After training, each model was given a starting prompt text to predict the next 20 characters. The models with more hidden states displayed a noticeable improvement in the sophistication of their predictions compared to the models with fewer hidden states. The table below shows the predictions of each GRU model from the text prompt.

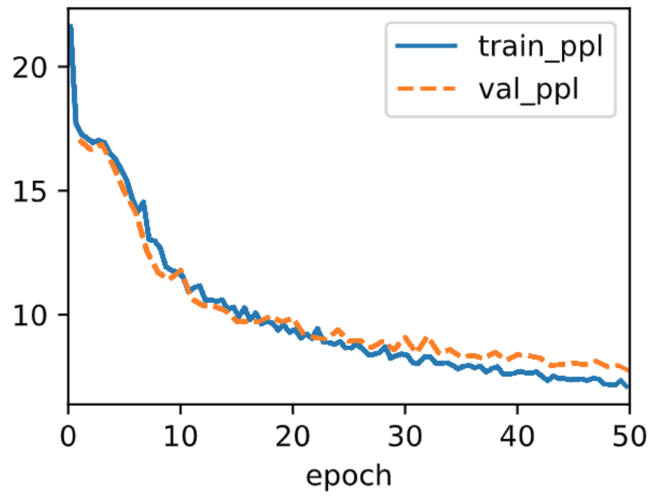
Model	Prediction from Text Prompt		
	‘it has...’	‘we are always...’	‘he looked across...’
GRU-8	the the the the the	the the the the the	the the the the the
GRU-16	mong the the the th	the the the the the	ing the the the the
GRU-32	the the the the the	ed the time the time	the time the time t
GRU-64	in thing that that	dimension in i sume	the psychologist yo
GRU-128	it and the time tra	the time traveller	ed the provention is

Problem 1b

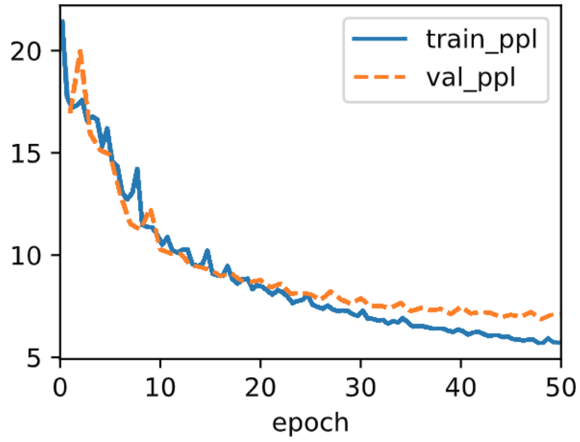
LSTM-8 state



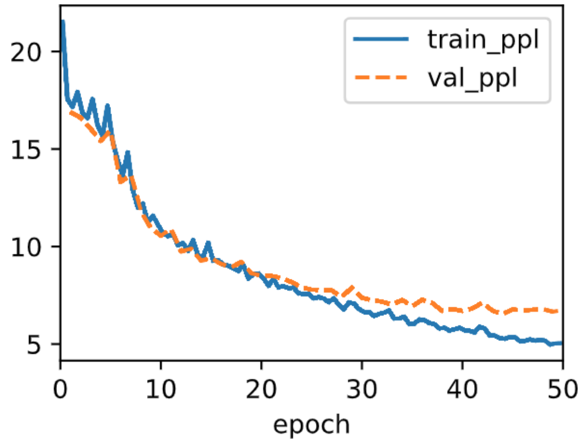
LSTM-16 state



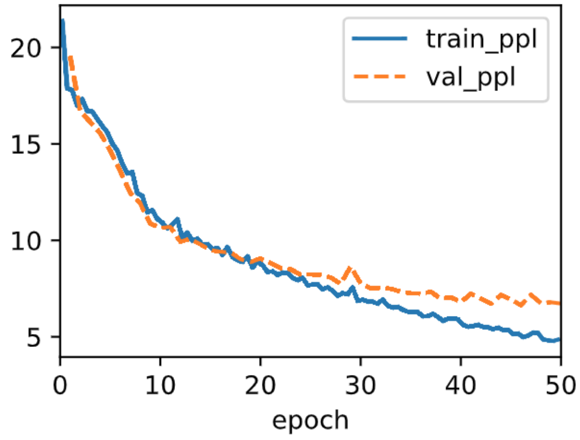
LSTM-32 state



LSTM-64 state



LSTM-128 state



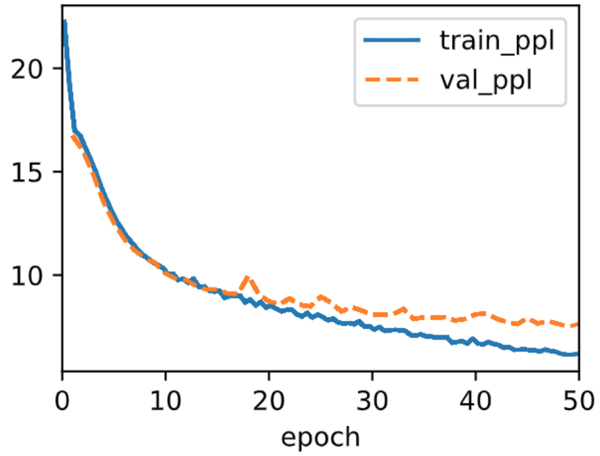
The trained LSTM models, like the GRU models in the previous section, demonstrated a steady increase in performance as more hidden states were incorporated. However, the LSTM models showed poorer performance in every metric measured when compared to their GRU counterparts. This was expected, as LSTM networks are generally less efficient than GRUs and thus require more parameters and training time. The table below compares the performance metrics of each of the 5 models trained, with a lower perplexity rank indicating lower perplexity.

Model	Perplexity Rank	Train Time	Size (Params)	Complexity (MACs)
LSTM-8	5	2min	1.22k	1.33M
LSTM-16	4	2min 3s	2.94k	3.18M
LSTM-32	3	2min 30s	7.94k	8.45M
LSTM-64	2	3min 54s	24.06k	25.3M
LSTM-128	1	7min 32s	80.9k	84.15M

After training, each model was given a starting prompt text to predict the next 20 characters from. The models with more hidden states displayed a noticeable improvement in the sophistication of their predictions compared to the models with fewer hidden states, though the predictions were generally not nearly as sophisticated as those of the GRU models from the previous section. The table below shows the predictions of each LSTM model from the text prompt.

Model	Prediction from Text Prompt		
	‘it has...’	**‘we are always...’**	**‘he looked across...’**
LSTM-8	the the the the the	and and and and and	and and and and and
LSTM-16	the the the the the	of the the the the	of the the the the
LSTM-32	the traveller and t	of the time travell	ions and the time tr
LSTM-64	inour and wather an	of and watheread h	have and the time t
LSTM-128	in and in and the t	er and the the grome	of the thing the ti

RNN-64 state



An RNN model with 64 hidden states was trained using the same dataset and epochs as the models in previous sections. Though the RNN-64 model didn't perform nearly as well in perplexity as GRU-64 and LSTM-64, it took less time to train and was considerably smaller in size and complexity than these two, which was expected as RNNs are much more simple compared to GRUs and LSTMs. The table below compares the performance metrics of each of the 3 models trained, with a lower perplexity rank indicating lower perplexity.

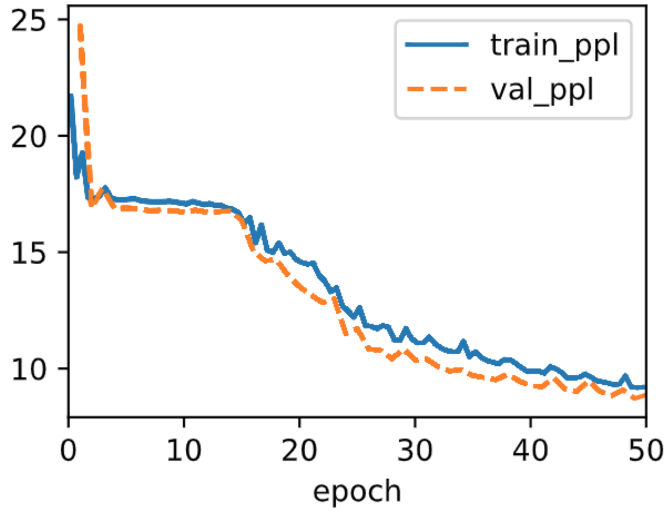
Model	Perplexity Rank	Train Time	Size (Params)	Complexity (MACs)
RNN-64	3	2min 11s	6.02k	6.23M
GRU-64	1	2min 43s	18.05k	18.94M
LSTM-64	2	3min 54s	24.06k	25.3M

After training, the RNN model was given a starting prompt text to predict the next 20 characters from. As expected, the RNN model didn't appear to attain outputs quite as sophisticated as the GRU and LSTM models. The table below shows the predictions of each 64 state model from the text prompt.

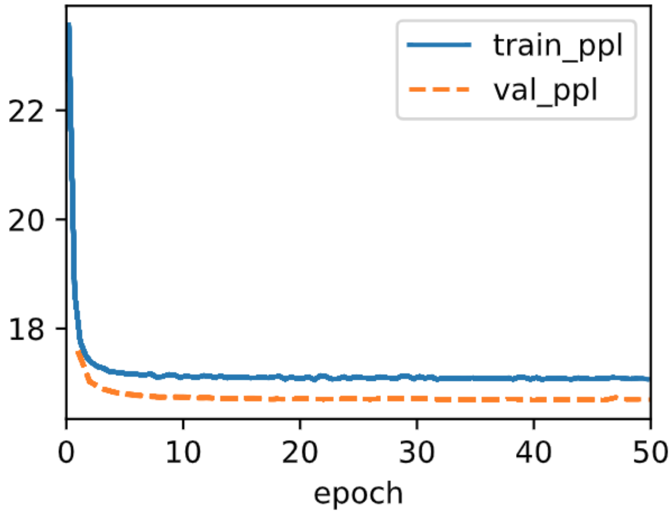
Model	**Prediction from Text Prompt**		
	'it has...'	**'we are always...'**	**'he looked across...'**
RNN-64	and hin the proun t	all has in the time	his said the time t
GRU-64	in thing that that	dimension in i sume	the psychologist yo
LSTM-64	inour and wather an	of and wathereard h	have and the time t

Problem 2

3-layer DeepGRU

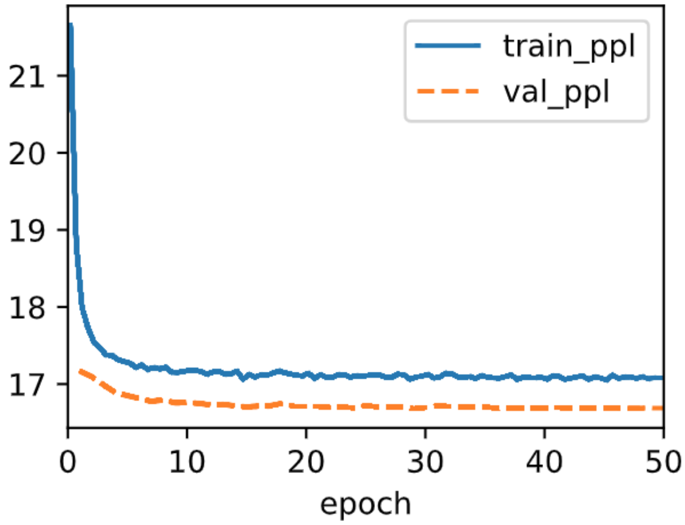


3-layer DeepLSTM

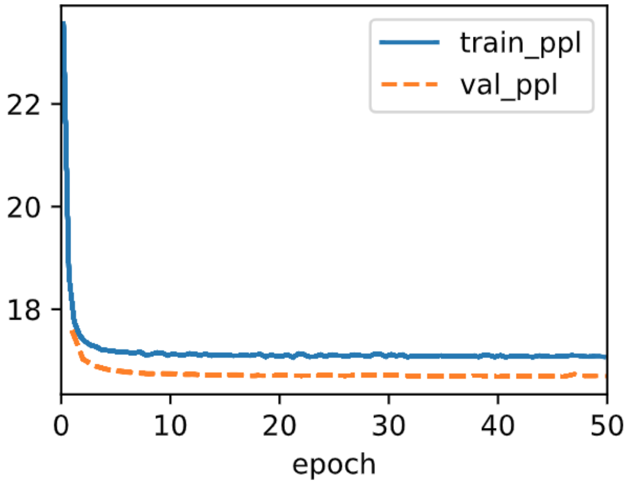


A deep GRU and deep LSTM model were created, each containing 3 layers, 32 hidden states, and dropout set to a value of 0.5. As seen in the plots above, the deep GRU model showed significantly better performance while the deep LSTM model appeared to stall in training. Naturally, the deep models also showed a considerable increase in training time compared to their single-layer counterparts.

5-layer DeepGRU



5-layer DeepLSTM



This process was then repeated for deep GRU and deep LSTM models containing 5 layers. After making this adjustment, it was clear that each of the network types could only handle so many layers until reaching a stalling point, with 5 layers appearing to cross that point. Even then, however, the deep GRU model appeared to outperform the deep LSTM model, reaching a validation perplexity value of approximately 16.7 compared to approximately 17.5 for the deep LSTM. The table below compares the performance metrics of each of the 4 models trained, with a lower perplexity rank indicating lower perplexity.

Model	**Perplexity Rank**	**Train Time**	**Size (Params)**	**Complexity (MACs)**
3-GRU	1	3min 14sec	18.62k	19.76M
3-LSTM	3 (roughly tied)	4min 33sec	24.83k	26.41M
5-GRU	2	5min 30sec	31.3k	33.19M
5-LSTM	3 (roughly tied)	6min 37sec	41.73k	44.37M

After training, each deep model was given a starting prompt text to predict the next 20 characters from. Though the 3-layer GRU did show steady convergence during training, its output prediction was very basic without much attained knowledge to demonstrate.

The stalling during training that the other models experienced appeared to greatly affect their inferences, with all three models simply outputting blank strings. The table below shows the predictions of each deep model from the text prompt.

Model	**Prediction from Text**		
	‘it has...’	**‘we are always...’**	**‘he looked across...’**
3-GRU	the the the the the	the the the the the	the the the the the
3-LSTM	‘ ‘	‘ ‘	‘ ‘
5-GRU	‘ ‘	‘ ‘	‘ ‘
5-LSTM	‘ ‘	‘ ‘	‘ ‘