1. **[40 pts]** Consider a Hidden Markov Model parametrized by \( \text{Pr}(y_t|y_{t-1}) \) and \( \text{Pr}(x_t|y_t) \). It satisfies the Markov property since the current state \( y_t \) depends only on the previous state \( y_{t-1} \). In practice, the current state may depend on earlier states and therefore the Markov property is not satisfied. It turns out that it is possible to ensure that the Markov property holds by augmenting the set of states.

(a) **[20 pts]** Show how to rewrite a process parametrized by \( \text{Pr}(y_t'|y_{t-1}', y_{t-2}) \) and \( \text{Pr}(x_t'|y_t') \) into an HMM parametrized by \( \text{Pr}(y'_t|y'_{t-1}) \) and \( \text{Pr}(x_t'|y'_t) \).

Let \( y' = (y_t, y_{t-1}) \), then

\[
\text{Pr}(y'_t|y'_{t-1}) = \text{Pr}(y_t, y_{t-1}|y_{t-1}, y_{t-2})
\]

(1)

where

\[
\text{Pr}(y_t=a, y_{t-1}=b|y_{t-1}=c, y_{t-2}=d) = \begin{cases} 
\text{Pr}(y_t|y_{t-1}, y_{t-2}) & \text{if } b = c \\
0 & \text{otherwise}
\end{cases}
\]

(2)

Also,

\[
\text{Pr}(x_t|y'_t) = \text{Pr}(x_t|y_t, y_{t-1}) = \text{Pr}(x_t|y_t)
\]

since \( x_t \) is conditionaly independent of \( y_{t-1} \) given \( y_t \).

(b) **[20 pts]** Is it possible to do this reparametrization without increasing the number of parameters? Justify your answer by counting the number of parameters before and after the transformation.

Let \( N \) be the size of the domain of \( y_t \) and \( M \) be the size of the domain of \( x_t \). Before the transformation, we need \( N^2(N-1) \) parameters to represent \( \text{Pr}(y_t|y_{t-1}, y_{t-2}) \) and \( N(M-1) \) parameters to represent \( \text{Pr}(x_t|y_t) \). After the transformation, a naive representation for \( \text{Pr}(y'_t|y'_{t-1}) \) would require \( N^2(N^2-1) \) parameters. However, all parameters that are set to 0 in Eq. 2 can be ignored. This leaves \( N^2(N-1) \) parameters. Similarly, after the transformation, a naive representation for \( \text{Pr}(x_t'|y'_t) \) would require \( N^2(M-1) \) parameters. However, since \( x_t \) is conditionally independent of \( y_{t-1} \) given \( y_t \), we can ignore \( y_{t-1} \) which leaves \( N(M-1) \) parameters. Hence we can do the reparameterization without increasing the number of parameters.
2. [60 pts]

(a) [20 pts] In theory LSTM and GRU units should perform better than linear units since they are more expressive than linear units and they can control the content of their memory by utilizing gates. However, the graph shows that linear units perform better. Since linear units have fewer parameters they need less data, which may explain why they need fewer iterations to find a good model. Similarly, GRU units have fewer parameters than LSTM units and therefore they need less data to find a good model. Another possible explanation is that the optimization of GRU and LSTM units is not as easy as Linear units and therefore they might be more prone to local optima.

![Graph showing performance comparison between Linear, LSTM, and GRU units](image)

(b) [20 pts] In theory, it should be sufficient for the RNN to know the category at the start without needing to receive the category as input at every step since it should be able to remember the category. Similarly, in theory, the RNN should not need to receive as input the previous character that it emitted since its hidden state contains the information of the previously emitted character. However, in practice, the RNN would need to learn to remember the category and previous character, which is a difficult task to learn. Hence feeding the category and the previous character as input at every step simplifies the job of the RNN and this regime yields the best results. Since the previous character is much more
informative than the category to determine the next character that should be emitted, feeding the previous character only at each step produces better results than feeding only the category at each step. The worse results are obtained when neither the category nor the previous character are fed as input at each step.

![Generating names with a character-level RNN](image)

(c) [20 pts] In theory, the results should be better with attention since the neural network can better align its outputs with the inputs even when the order of the words changes. However, the model without attention has fewer parameters and therefore slightly less data is needed to obtain a good translation in this dataset.