ure 11: Recurrent Neural Networks

★ Slide-10 Some terminology

- You can imagine negative of the loss as the 'reward' in supervised learning.
- In Reinforcement Learning, you basically learn by doing. You explore & exploit things a lot to get a handle on the process.
- Reinforcement learning is ~~used in~~ analogous to Games where you play it a lot of times with mistakes & new discoveries to get good at the game.
- Model based R-L :- where you have a defined Transition & Reward function.
- Model-free RL :- where you don't have T & R functions, but you estimate them using techniques like Q-Learning.

★ Slide - 30 Text prediction: supervised Learning

- For ease of use, we assume that only letters & spaces will be used, thus, we will have 26 + 1 = 27 classes for classification. The next character for the prediction will be the label of the classification.

- One idea on how to featurize the text data would be to use all the previous characters before the label as features (like shown in the table)
- But that's an issue. Encoding it won't be a problem. But it doesn't have a fixed dimension. It's dimension increases as we go along the string of input text.

- One more idea would be to use only 1 single previous character. But, you can see that means we will loose some info. Because predicting a character based on single previous character is meaningless because they don't follow any pattern. ~~Words in the sentence follow the pattern.~~ Letters/characters in a word follow a pattern. So, we need more than 1 character ~~as req~~ for label prediction.

- So, a better idea is to use 'm' previous characters, where $m > 1$ but we don't know what m is yet. We have to decide that.

★ Slide - 67 Can express as a state machine

- One example of the expression of this text as state machine: Assume 'S' is a state which comprises of 'm=3' latest characters in the string of text. So set of all possible states is all ordered m=3 characters.

- ~~Set of~~ An input or action to this ~~MDP~~ state machine is a single character that we append to the state's string which in turns updates the state. So, set of all possible inputs/actions is set of all character.

- Initial state: We use the special character '^' as a placeholder for the initial state. For example, assume that you've unlocked your phone & went to a texting app. You haven't typed anything yet, but we would still like to predict what you'll type. Hence we use '^^^' as our

initial state where '^' isn't the only character you could use (obviously). It's a choice.
Also, we add '^' to the beginning of a new string data set as you can see in $x^{(1)}$, $x^{(2)}$ & $x^{(3)}$.

- Transition function $f(s,x)$ just updates the last 'm' characters with 'x' action.

- Now, for output, we could use something stochastic like softmax function. So, the output will be a vector of probabilities of all characters.

- The final output function $g(s)$ takes input state 's' & uses the softmax output to determine a label. This $g(s)$ is our multi-class linear classifier.

★ Slide - 124 Can express as a state machine

- Here, $S_0, S_1, S_2, \ldots$ are the states. $x_1, x_2, x_3 \ldots$ are the character inputs / actions ~~for~~ from a string $x^{(1)}$. 'f' is the transition function. 'g' is the output function 'p' is the output vector of probabilities from 'g'.

- S_t has size $m \times 1$. It's a vector consisting of 'm' latest characters. x_t is 1×1 character. p_t is $V \times 1$ vector where V may the the total no. of possible characters. Incase of $V = \{A, b, c, d \ldots, z, \wedge, \_\}$ then $V = 28 \times 1$. In case $V = \{0, 1\}$, then $V = 2 \times 1$

- Now, we wanna express transition function f mathematically $s_t = f(s_{t-1}, x_t)$ i.e. current state is function of previous state & action taken / input added. Now, we can express this equation mathematically using matrix multiplication.

- We wanna choose the multipliers of s_{t-1} in such way that the bottom element gets out; remaining two elements are shifted downwards by 1 & top element becomes zero.
- We wanna choose the multiplier of x_t such that only top element has x_t & rest elements are zero, ~~So that when added not multiplied~~. The matrices given in the slide ensure that.

- Now, $p_t = g(s_t) = f_2(W^o s_t + W_0^o)$ i.e. we use v-class logistic regression. Now we only have to choose f_2 & learn W^o & W_0^o.