RNN
(Vanilla + LSTM)

DSC 180B - Group 3 - Week 5

Vanilla RNN - Overview

- **Task**: Predict # *hours* an app is used during *a day*
  - **Library**: Pytorch
  - **Input**: one-hot encoding: Monday → Sunday
    - Monday: [1,0,0,0,0,0,0]
    - Sunday: [0,0,0,0,0,0,1]

    } ⟶ X
  - **Output**: duration (in hrs) of using an app in a day → y
    - MinMaxScaler for the output → speed up training
  - **Train/Val/Test split**: 70/20/10, no shuffle

Vanilla RNN - Implementation

- **class VanillaRNN**
  - **__init__**:
    - init Vanilla RNN model (# nodes/layers, in/out dims, dropout rate)
  - **forward**:
    - input data
      → network (1 hidden layer, in forward direction)
      → fully connected output
- **class Optimization**
  - **__init__**
  - **train_step**
  - **train**
  - **evaluate**
  - **plot_losses**

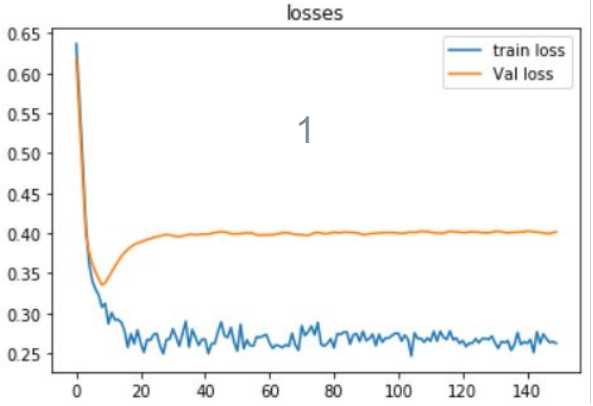
Vanilla RNN - Implementation

- **class Optimization**
  - **__init__**
    - Init everything related to training the model
      - Model (Vanilla RNN), loss func (L1 loss), optimizer (Adam)
      - Train/Val losses (empty np.arrays)
  - **train_step**
  - **train**
  - **evaluate**
  - **plot_losses**

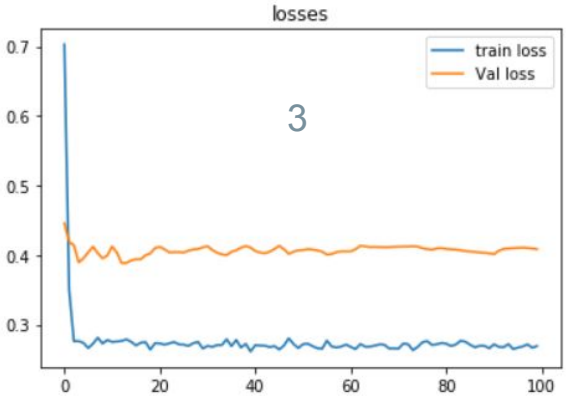
} funcs with self-explanatory names

Vanilla RNN - Results

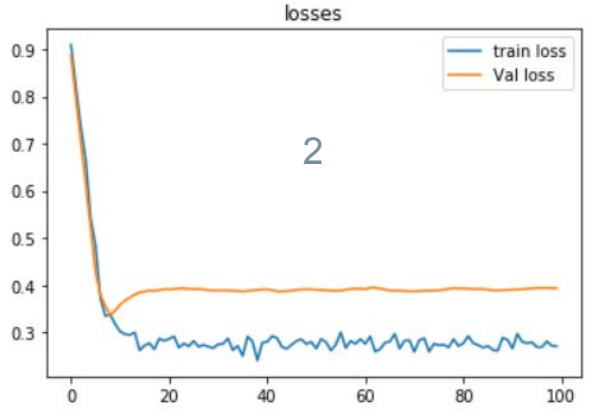
Exp	# layers	# nodes	batch size	n_epochs	dropout	lr
1	5	5	7	150	0.1	1e-3
2	5	5	7	**100**	0.1	1e-3
3	5	5	7	**100**	0.1	**1e-2**
4	**6**	5	7	150	0.1	1e-3



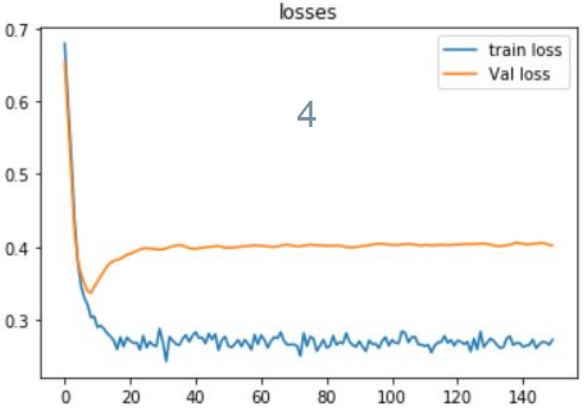
	test_val	test_pred
0	0.000000	0.539957
1	0.344352	0.547945
2	1.000000	0.539792
3	0.450793	0.540033
4	0.429142	0.539957
5	0.073159	0.539494
6	0.672383	0.538483
7	0.021944	0.540277



	test_val	test_pred
0	0.000000	0.366450
1	0.344352	0.344456
2	1.000000	0.357834
3	0.450793	0.362495
4	0.429142	0.366450
5	0.073159	0.362662
6	0.672383	0.355648
7	0.021944	0.356238



	test_val	test_pred
0	0.000000	0.213696
1	0.344352	0.208320
2	1.000000	0.209849
3	0.450793	0.210502
4	0.429142	0.213696
5	0.073159	0.208956
6	0.672383	0.209943
7	0.021944	0.209748



	test_val	test_pred
0	0.000000	0.295590
1	0.344352	0.295476
2	1.000000	0.295954
3	0.450793	0.296023
4	0.429142	0.295590
5	0.073159	0.295514
6	0.672383	0.295202
7	0.021944	0.296053

Vanilla RNN - Results

LSTM - Keras

Demo