

# **task\_9.1**

## **Machine Learning (WiSe 2025/2026)**

Author: Suvansh Shukla  
Matriculation No. 256245

---

### **Assignment 9 Task 1**

#### **Part A**

One of the reasons why such algorithms like Naive Bayes might fail is because it assumes that there is no correlation between any of the instances in the dataset. We know this to not be true in real life, especially in the case of time-series where one instance is dependent on its predecessor.

Another case where a non-sequential data classification algorithm like K Nearest Neighbors may fail is that the proximity or similarity of data may not have any actual relevance. This would be especially obvious when trying to use KNN for text prediction, just because two words are closer to each other in proximity doesn't mean that one will always come after another.

#### **Part B**

Feed-Forward Neural Networks process data in two ways: batch-wise or stochastically. In batch-wise processing the Neural Network doesn't discriminate among the inputs, and treats them all as discrete inputs with no connection. This means that if we tried to process text "The fruit basket" the neural network would process all the words but loose all the information that is present in their order. Now in stochastic (or one-by-one processing) the moment the network started to process "fruit" it would have forgotten all about "the" thus loosing critical information again. It's like the network has no memory.

Recurrent Neural Networks are able to retain things in memory and use it when processing succeeding data. So when a Recurrent neural network tries to process "fruit" it remembers that "the" came before it. This allows the recurrent neural network to capture the information inside the order of occurrence of instances.

#### **Part C**

Backpropagation through time requires a concept called "unrolling". Unrolling is equivalent to creating a copy of the RNN for each timestep, treating the recurrent network as a very deep feed-forward network where

each "layer" corresponds to a timestep. An interesting thing to note is that unlike the feed-forward networks the weights for each input is the same and it is just the hidden state that is propagated to through to the next iteration, carrying information from the past to the future.

The "Vanishing Gradient" problem still exists here, unfortunately. Because the weights are still updated by multiplication (just across timesteps now) when flowing backwards, they are still susceptible to diminishing to very small values. This means that early timesteps don't get any meaningful adjustment. Similarly, due to multiplication the adjustment values may increase exponentially causing the "Exploding Gradient" problem as well. [1]

## Part D

LSTMs or Long-Short Term Memory helps solve the problem of "Vanishing Gradient" by introducing three gates: "input gate", "forget gate" and "output gate".

- Forget Gate: decides which information to forget from the previous hidden state and the current input. It uses a Sigmoid activation function.
- Input Gate: decides which new details should be added to the cell state. It has two activation functions, a sigmoid function that selects values to update the cell and a tanh activation function that produces new candidate values for the cell state.
- Output Gate: defines what data should be presented as the LSTM cell's current output. It processes the updated cell state and the present hidden state to produce the output.

Though this helps address the "Vanishing Gradient" problem, it does not meaningfully solve the "Exploding Gradient" problem. [2]

## Part E

- Classifiers that work on non-sequential data fail on sequential datasets because they assume that data points have no correlation with each other across time. This is not always the case.
- Neural networks maybe modified to handle and process sequential data by saving past states across time and using them to calculate the output for the next input.
- Backpropagation in recurrent neural networks involves "unrolling" through time to allow the hidden state to propagate past information to the future for use.
- Recurrent Neural Networks are still susceptible to "Vanishing" and "Exploding" Gradient problems, though Vanishing Gradient problem may be addressed by LSTMs, that use gates for selective data propagation and retention.

- 
1. <https://mbrenndoerfer.com/writing/backpropagation-through-time-rnn-training-algorithm> ↵
  2. <https://www.baeldung.com/cs/lstm-vanishing-gradient-prevention> ↵