Neural Networks Homework 11

Sayeh Jarollahi (7073520, saja00006@stud.uni-saarland.de)
Mahsa Amani (7064006, maam00002@stud.uni-saarland.de)

January 25, 2025

Exercise 10.1

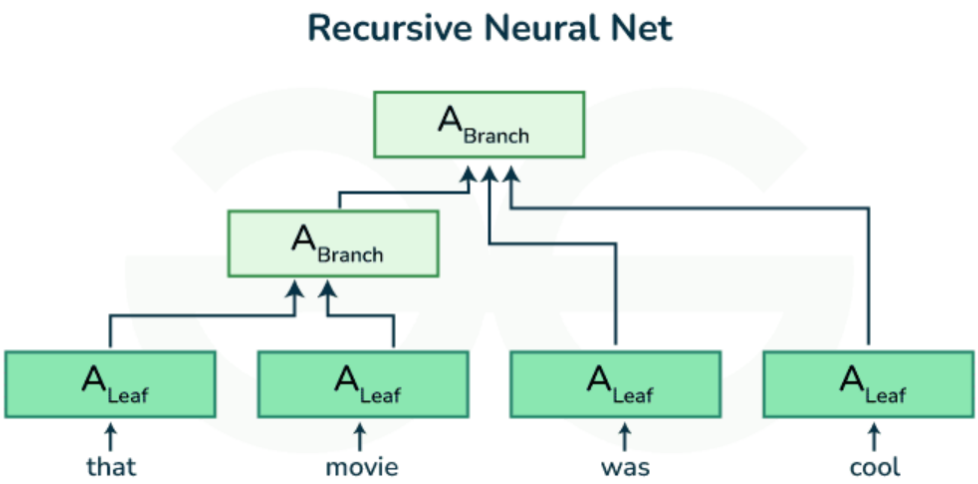
Proof. 1. The key difference between CNN and RNN lies in the type of data they model. CNNs are designed to handle spatial data, such as images, by applying convolutional filters to capture local patterns. They assume independence between input features and process inputs in parallel. RNNs, on the other hand, are designed for sequential data, such as time series or text, by maintaining a hidden state that captures temporal dependencies. This enables RNNs to model relationships between data points over time. Reference Material: RNN, LSTM, and GRU
2. Training RNNs often suffers from the vanishing or exploding gradient problem due to repeated multiplication of gradients across timesteps. This is exacerbated when using activation functions like `Tanh` or logistic sigmoid, as their derivatives tend to saturate. LSTMs mitigate this by introducing gates (input, forget, and output gates) and a cell state, which allows information to flow without being entirely overwritten. This structure helps preserve gradients over long sequences. LSTMs share similarities with residual networks (ResNet) because both architectures address vanishing gradients by creating pathways for gradient flow, though their mechanisms differ. Reference Material: LSTM and ResNet

□

Exercise 10.2

Proof. 1) **One to One**: Image classification. The input is an image (for example MNIST images) and the output is a single label regarding the input.
One to Many: It can be image captioning in which a single image is given in the input and the output is a sequence of words related to that image.
Many to One: It can be text classification, to be more specific Sentiment Analysis. In this case a sequence of words are given in the input and in the output it says if the sentence is positive or negative.
Many to Many, 4th: It can be Machine Translation. The input is a sequence of words in one language and the output is the exact sentence in another language.
Many to Many, 5th: In this case we can say video classification with frame-wise labels. The inputs are the frames of a video and the output is a sequence of labels for each frame.

2) We can show it as follows:



Reference: GeeksforGeeks
Recursive NNs have a tree-like structure where inputs (e.g., words in a sentence) reside at the leaf nodes, and their representations are combined as they propagate upward through hidden units. The network produces a final representation or prediction at the root node, typically summarizing the hierarchical input (e.g., determining the sentiment of a sentence). Recursive NNs are often used in natural language processing tasks, such as syntax tree parsing or sentiment analysis, where the hierarchical relationships between elements are crucial.
3) a) The term J_{k+1} represents the Jacobian matrix of $\mathbf{h}_{k+1}$ with respect to $\mathbf{h}_k$. The Jacobian J_{k+1} is given as:

$$J_{k+1} = \frac{\partial \mathbf{h}_{k+1}}{\partial \mathbf{h}_k} = \text{diag}(\sigma'(\mathbf{z}_{k+1})) \cdot \mathbf{W}$$

where $\text{diag}(\sigma'(\mathbf{z}_{k+1}))$ is a diagonal matrix where each diagonal element is the derivative of the activation function σ evaluated at $\mathbf{z}_{k+1}$, $\mathbf{W}$ is the weight matrix for the recurrent connections.
b) The two issues that can occur are: **Vanishing Gradients:** If the spectral norm of $\mathbf{W}$ and $\sigma'(\mathbf{z}_t)$ is less than 1, the product

$$\prod_{k=t}^{T-1} \|J_{k+1}\|$$

diminishes exponentially as T increases. This results in gradients approaching zero, making it difficult for the model to learn long-term dependencies. **Exploding Gradients:** If the spectral norm of $\mathbf{W}$ and $\sigma'(\mathbf{z}_t)$ is greater than 1, the product

$$\prod_{k=t}^{T-1} \|J_{k+1}\|$$

grows exponentially, causing gradients to become excessively large. This can lead to instability during training.

Both issues stem from: The magnitude of $\mathbf{W}$, specifically its spectral norm $\|\mathbf{W}\|$. Also, the derivative of the activation function $\sigma'(\mathbf{z}_t)$, which depends on the choice of the activation function.

□