2/4/2024

Student Name: Ryan Mckee

Student number: 40294886

Assignment 2

CSC3066 Deep Learning (Fake News Detection)

Introduction

In an era marked by the rapid dissemination of information on social media platforms like Twitter, our assignment focuses on developing machine learning models to automatically detect fake news within Twitter posts. As data scientists, our objective is to provide our client with a scalable and accurate solution to mitigate the risks associated with the spread of misinformation. By leveraging artificial neural networks (ANN) and word embedding models, we aim to equip our client with a robust system capable of classifying tweets as either genuine or false. This report documents our journey through implementing and evaluating various ANN architectures, including Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) models, exploring techniques and settings to enhance model performance and provide actionable insights to combat the proliferation of fake news effectively.

Baseline models

Baseline models performed relatively well, for each model a set of baseline hyperparameters were set and no pre-processing on the data was performed for the purpose of providing a skeleton model for better analysis of the performance of the models and allow for looking into what techniques could be used to

Baseline hyperparameters-for all models

Reason for hyperparameters. Batch size 128 is a default hyperparameter size used for most models. It’s big enough that the model will train fast, 10 epochs is enough to see the the stats over that time. Optimizer is Adam this optimizer is an improved version of gradient descent therefore making it safe to assume this is the best optimezer for our model

0.05 learning rate default value to start with. Validation split is 10% of the training data therefore 200 of the 2000 samples. Loss uses binary crossentropy due to this being the most effective loss function for the binary classification problem I am trying to implement. Each model will start off with one hidden layer 5 neurons and leaky relu as I expect dead neuron problem to occur with normal relu due to the padding of the training and testing data giving 0 dimensional values for entire tokens within the samples and potential negative dimensional totals for certain values.

# Hyperparameters

batch\_size = 128 # Baseline batch size

epochs = 10 # Baseline number of epochs

learning\_rate = 0.05 # Baseline learning rate

optimizer = tf.keras.optimizers.Adam(learning\_rate=learning\_rate) # Baseline optimizer

validation\_split = 0.1 # Baseline validation split: 10% of the training data

activation = 'leaky\_relu' # Baseline activation function - leaky relu, used to prevent dying relu problem as minus numbers and zeros values are a possiblity for some inputs because of dimensional spaces of embedded dimensions

loss = 'binary\_crossentropy' # Baseline loss function:binary crossentropy used for binary classification model

hl1\_size = 5 # Baseline number of hidden layer neurons

Word Embedding vector

* Training accuracy and validation improve over epochs. While validation loss

A graph of a graph

Description automatically generated

Analysis and reporting of the results from Task One

Final outcome of project