



CSE-5368 Neural Networks

Exercise Problems 08



Complete the following function. This function should create and train an autoencoder using tensorflow.

```
import numpy as np
import tensorflow as tf
def autoencoder(train,test,encoder_layers,latent_space,epochs, alpha):
# train: numpy array of input [nof_train_samples,input_dimensions]
# test: numpy array of input [nof_test_samples,input_dimensions]
# encoder_layers: list of integers representing number of nodes in
# each layer. The last number represents the dimension of latent space
# return: output of the latent space, output of autoencoder ,
# Notes:
# Assume all layers are fully connected.
# Activation function for all the layers should be sigmoid.
# Use mse to calculate loss.
# Use gradient descent.
# Assume the decoder has the same layer structure as the encoder in the
reverse order.
```



CSE-5368 Neural Networks

Exercise Problems 08



Using tensorflow, complete the following function. This function should implement the forward pass of an RNN.

```
import numpy as np
import tensorflow as tf
def rnn_forward_pass(Wxh, Whh, Why, bh, by, input_sequence):
# input_sequence: numpy array [time_steps, embedded_dimension]
# return: output sequence, hidden state
```



CSE-5368 Neural Networks

Exercise Problems 08



Using tensorflow, complete the following function. This function should train an RNN

```
import numpy as np
import tensorflow as tf
def train_rnn(Wxh, Whh, Why, bh, by, input_sequence, targets, alpha,
epochs):
# input_sequence: numpy array [batch,time_steps,embedded_dimension]
# return: Wxh, Whh, Why, bh, by
# Notes:
# Use mse to calculate loss
# Use gradient descent
```