# Quiz 5: Introduction to Deep Learning

Introduction to Supervised Learning

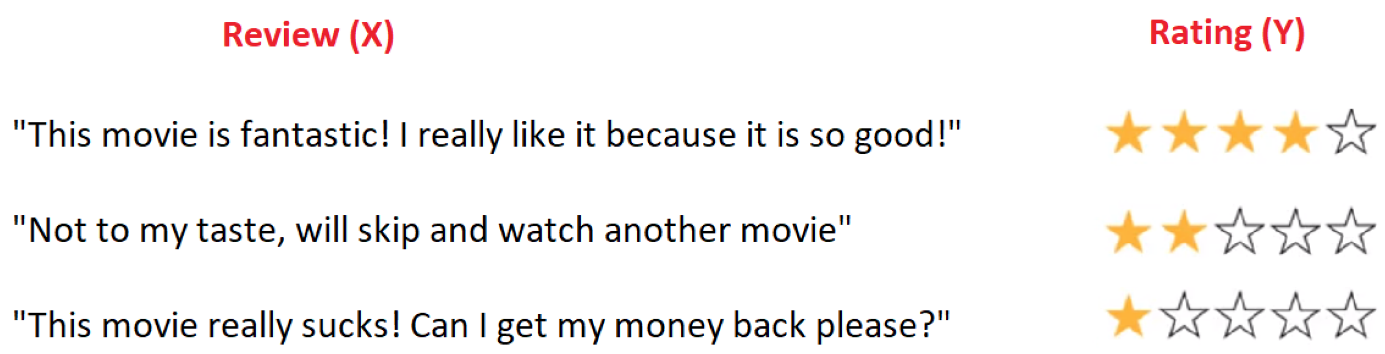
Email address *

mikesymmonds@gmail.com

Please enter your name: *

Michael Symmonds

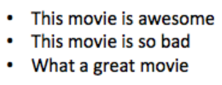
We want to classify movie reviews into 5 categories: 1 to 5 stars. (1 for the worst movies, 5 for the best movies)



| Review (X) | Rating (Y) |
|---|---|
| "This movie is fantastic! I really like it because it is so good!" | ★★★★☆ |
| "Not to my taste, will skip and watch another movie" | ★★☆☆☆ |
| "This movie really sucks! Can I get my money back please?" | ★☆☆☆☆ |

## Processing sequences of integers (a small example)

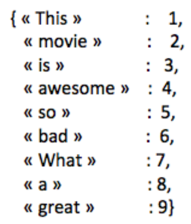
Consider the following documents:



- This movie is awesome
- This movie is so bad
- What a great movie

Using the following dictionary, how would the second document be encoded?

1 point



```
{ « This »      :  1,
  « movie »     :  2,
  « is »        :  3,
  « awesome »   :  4,
  « so »        :  5,
  « bad »       :  6,
  « What »      :  7,
  « a »         :  8,
  « great »     :  9}
```

○ [1, 2, 4, 5]

○ [7, 8, 10, 2]

● [1, 2, 3, 5, 6]

We want to use One Hot Encoding to transform the list of sequences into a tensor that we can feed to a neural network, what would be the shape of this tensor? 1 point

● (3, 9)

○ (9, 5)

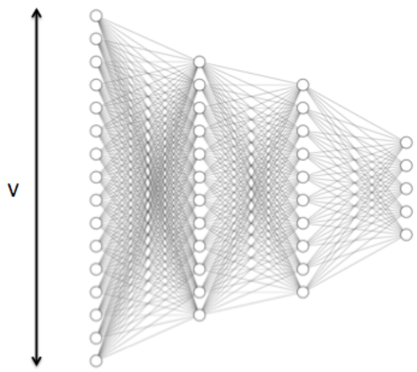
○ (3, 3)

What would be the first row of this tensor? 1 point

[1,1,1,1,0,0,0,0,0]

## Building the model

Now that the data has been preprocessed. We want to feed the tensor in a Deep Neural Network with several Dense layers.



How many neurons should the last layer contain? 1 point

○ 1

● 5

○ 10

What should be the activation function in the last layer? 1 point

● softmax

○ sigmoid

○ tanh

What should be the loss function? 1 point

○ Binary cross entropy

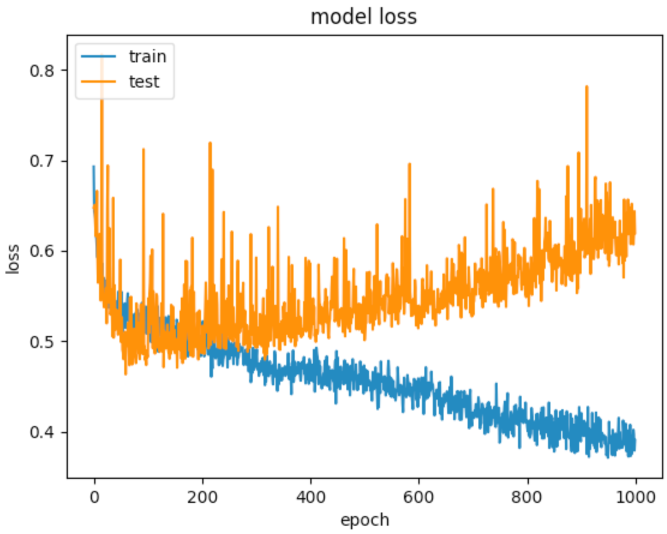
● Categorical cross entropy

○ MSE

How is this loss related to Maximum Likelihood Estimation? 1 point

This loss function is equal to the normalised negative log-likelihood function. It is the log-likelihood function, divided by N (the number of documents) and then multiplied by -1. This is so we can minimise the loss function (which is easier to do on a computer than maximisation). The log-likelihood function is equivalent to the likelihood function which, at it's maximum, maximises the probability that the process described by the model produced the data that were actually observed. i.e. determines the parameters that best suit the data in terms of likelihood.

After the training process, we obtain the following validation and training losses. What is the problem? 1 point



Here we can see the classical problem of overfitting. After the initial (around 200) epochs of training, we see that the loss function is decreasing for both the training and testing set. However, thereafter the loss function of the testing data starts to increase as the model as become too specific to the training set. Thus, as the loss function of the training set continues to decrease as the model fits that dataset better, the model starts to perform worse on the testing set as it is overfitted to the training data.

How can we solve the previous problem? 1 point

There are a few ways to solve the problem of overfitting in this case. These include:

Reducing the complexity of the model (ie decreasing the number of hidden layers or the neurons in each layer)
Applying the dropout technique (whereby you randomly exclude a number of output features of a layer/s in the training process)
Weight regularisation (whereby you alter the loss function to penalise having large weights)

Explain why the previous model is suboptimal regarding the nature of data 1 point

The key issue with the model we have used is that it ignores the sequentiality of the data. It treats each document as a bag of words and associates the sentiment of the document with the frequency of words that it associates with a good or bad score. Hence, it does not fully capture the sentiment of the document and may make mistakes in classification. A simple example is a review like "Everyone said this movie was terrible but I did not think so". The algorithm would see the word terrible and likely assign a low score. However, the review is clearly positive. Hence, using a classifier that incorporates the sequentiality of words would better handle the sentiment inherent in the reviews.

## Programming Session

Did you understand the problem?

● Yes

○ No

Feel free to send us an email if you need more support.

Google