Review

We wish to create a model to predict the stock movement based on financial news articles. The training data is composed of a list of news associated with the following targets: 1 if the daily return is > +1%, 0 if the daily return is between -1% and 1% and -1 if the daily return is < 1%

Preprocessing the data

The dataset is composed of N news of different lengths. We split the dataset into N_train samples for the training, N_validation samples for the validation, and N_test samples for the testing.

$$
\begin{array}{lllllll}
\text{Sentence 1} & w_1^1 & w_1^2 & w_1^3 & \dots & w_1^{T_1} & \\
\text{Sentence 2} & w_2^1 & w_2^2 & w_2^3 & \dots & w_2^{T_2} & \\
\text{Sentence 3} & w_3^1 & w_3^2 & w_3^3 & \dots & & w_3^{T_3} \\
 & \vdots & & \vdots & & \vdots & \\
\text{Sentence N} & w_N^1 & w_N^2 & w_N^3 & \dots & w_N^{T_N} &
\end{array}
$$

Describe the first preprocessing step, which consists in transforming the list of sentences into a list of lists of integers.

To transform the list of sentences into a list of integers, we need to specify the vocabulary size (i.e, the number of words we want to process). Then, we create the word index dictionary, which maps each of the V most frequent words in the corpus into a unique integer.
Once it's done, we transform each document (i.e, list of words) into a list of the corresponding integers via the word index dictionary.

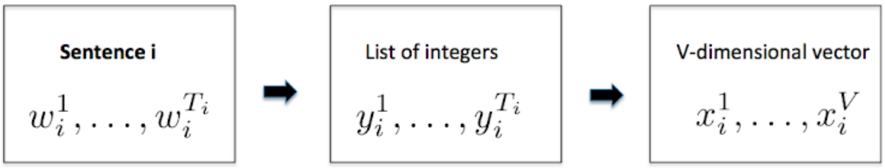
Explain briefly why we split the dataset into train-validation-test data instead of train-test data?

The models usually come with a certain number of hyperparameters. Finding the optimal model requires testing some (or all) the hyperparameters and keeping the set of hyperparameters which maximizes the chosen evaluation metric.

We use the validation data to evaluate the model for each set of hyperparameters. It results in information leaks (i.e, some information about the validation data leaks into the model).

Therefore, we should keep the test data as a never-before-seen data to evaluate the model.

Let V be the vocabulary size. How can each list of integers (representing a sentence) be encoded into a V-dimensional vector?



We use the one-hot encoding method. Each sequence of integers Y is represented by a vector of size V. All the dimensions of this vector are associated with zeros, except the ones corresponding to the integers that are present in Y, which are associated with ones.

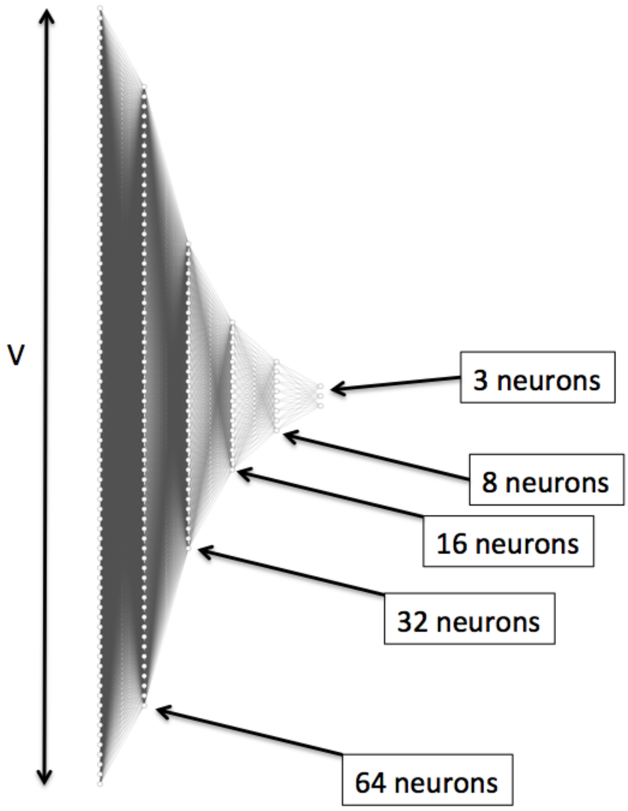
What would be the shape of the targets after the one-hot encoding process?

The shape of the targets for the training data is (N_train, K) where K = 3.
The shape of the targets for the validation data is (N_validation, K)
The shape of the targets for the testing data is (N_test, K)

A Feedforward Neural Network

We wish to use a feedforward neural network to classify the news using the following network:

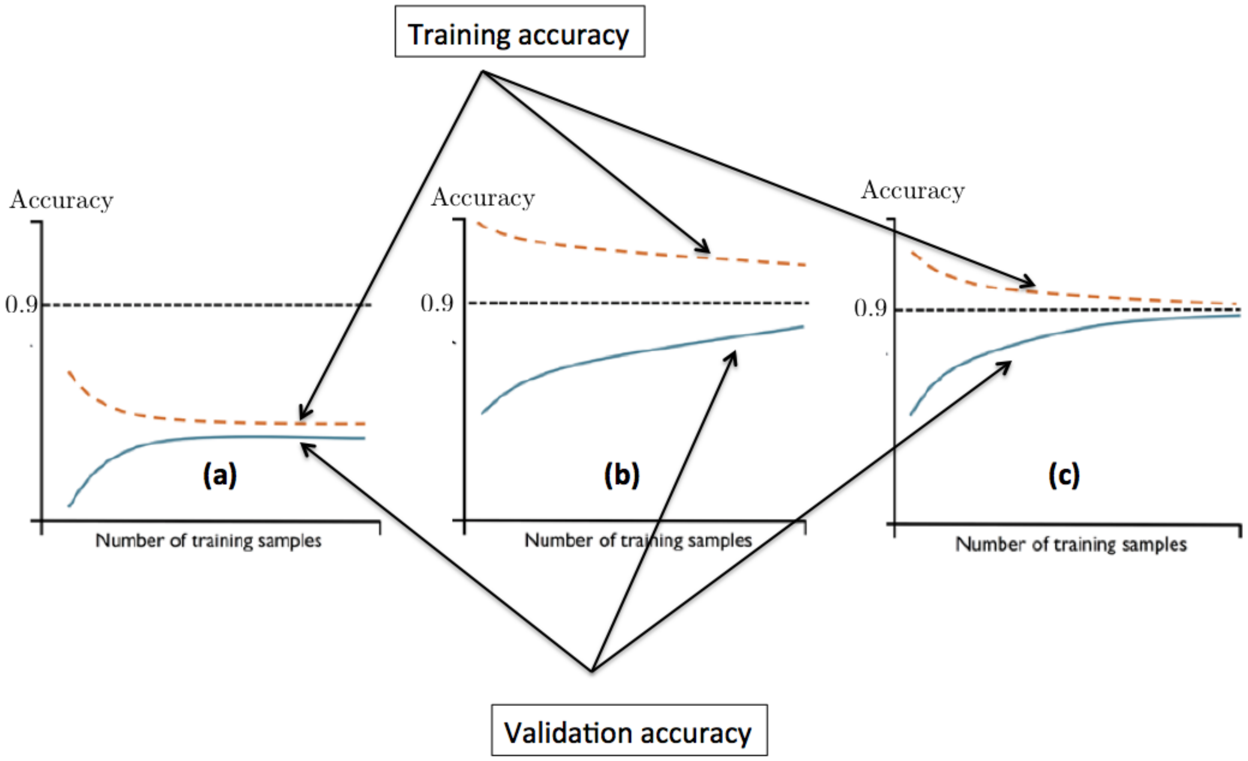
Let V = 20000. We use 5 dense layers of lengths: 64, 32, 16, 8 and 3.



Describe the evolution of the data shape after each layer transformation.

The input of the training data is of shape (N_train, V). After 5 dense layers, we have the following evolution of the shape of the training data:
(N_train, V) -> (N_train, 64) -> (N_train, 32) -> (N_train, 16) -> (N_train, 8) -> (N_train, 3)
For the validation data, we can replace N_train by N_validation.
For the test data, we can replace N_train by N_test.

We wish to compare 3 different architectures: model (a), model (b), and model (c). To that end, we plot the training and validation accuracy with respect to the number of samples for each model, we obtain the following curves:



Which model would you prefer and what issues do the other models exhibit?

Model (a) suffers from a high bias problem. Both the training and the validation accuracy converge to the same value, but this accuracy value is not that high. It's clearly an underfitting problem that requires increasing the model complexity.
Model (b) suffers from a high variance problem. There is a gap between the training and the validation accuracy. It's clearly an overfitting problem. We should reduce the model complexity, add some regularization or add even more samples.
Model (c) is the best one. Bost the training and the validation accuracy converge to the same value, which is a good accuracy. Model (c) represents the optimal model complexity.