Machine Learning and Finance
Final Exam - Session 1 - (2 hours)

The exam is composed of three independent problems:

- **Credit Risk Prediction** (40 marks)
- **Building Context-Based Embedding Vectors** (35 marks)
- **A Sequential Neural Network** (25 marks)

1 Credit Risk Prediction [40 marks]

We wish to create a model to assess the quality of a loan. We build a machine learning algorithm that predicts how likely the loan will be paid based on several features. There are three different possible labels:

- **Category A** if the likelihood of paying the loan is very high.
- **Category B** if the likelihood is neither high nor low.
- **Category C** if the likelihood is very low.

Convention: The category A is mapped to the index 0, the category B is mapped to the index 1 and the category C is mapped to the index 2.

To train the models, we use a training dataset composed of N samples, each sample is a vector of size $D = 20$.

Let X be the training input tensor containing all the samples. X is then of shape (N, D). Let T be the tensor of the targets after the one hot encoding process.

As shown in the Figure 1, the model we want to use is a neural network composed of one input layer with D passthrough neurons, followed by one hidden layer with $M = 10$ neurons, and finally one output layer with $K = 3$ neurons.

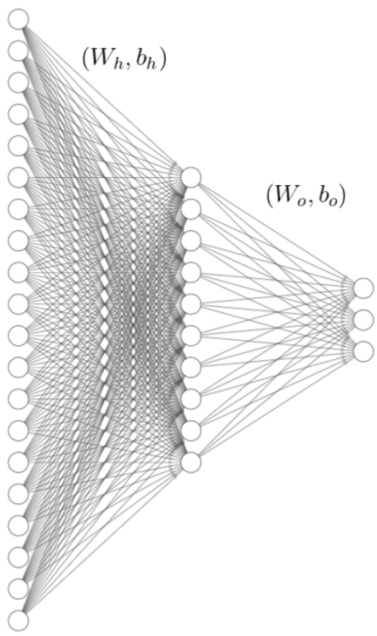

Figure 1: The Shallow Neural Network

We split the training input tensor X into several batches of size N_b. Let $\tilde{X}$ be the part of the training ïnput tensor containing the first N_b samples and $\tilde{T}$ the corresonding target tensor of shape (N_b, K).

- **Q1: Name three popular activations functions and draw them.** *[3 marks]*
- **Q2: Which activation function would you use for the last layer. Justify your answer.** *[3 marks]*
- **Q3: What are the shapes of the hidden layer's weight vector W_h and its bias vector b_h?** *[3 marks]*
- **Q4: What are the shapes of the output layer's weight vector W_o and its bias vector b_o?** *[3 marks]*
- **Q5: What is the shape of the network's output matrix P if we perform the forward propagation on $\tilde{X}$** *[3 marks]*
- **Q6: Write the equation that computes the network's output matrix P as a function of $\tilde{X}, W_h, b_h, W_o, b_o$.** *[4 marks]*
- **Q7: Write the loss function associated with this classification problem for the batch $\tilde{X}$ as a function of $P, \tilde{T}$** *[4 marks]*
- **Q8: What is backpropagation and how does it work?** *[5 marks]*
- **Q9: List all the hyperparameters you can tweak in this model?** *[6 marks]*
- **Q10: If the model overfits the training data, how could you solve the problem? (List three methods)** *[6 marks]*

2 Building Context-Based Embedding Vectors [35 marks]

2.1 A Context-free embedding model

2.1.1 The Word2vec/GloVe models

Q11: Describe the process of getting the embedding vectors using one of the two following models: Word2vec or GloVe. Make sure to specify the following elements: *[6 marks]*

- **How to prepare the dataset from a large corpus**
- **How to train the model**
- **How to extract the trained embedding vectors**

2.1.2 Using pre-trained embedding vectors in a classification problem

Consider the problem of predicting the next word using the architecture in Figure 2.

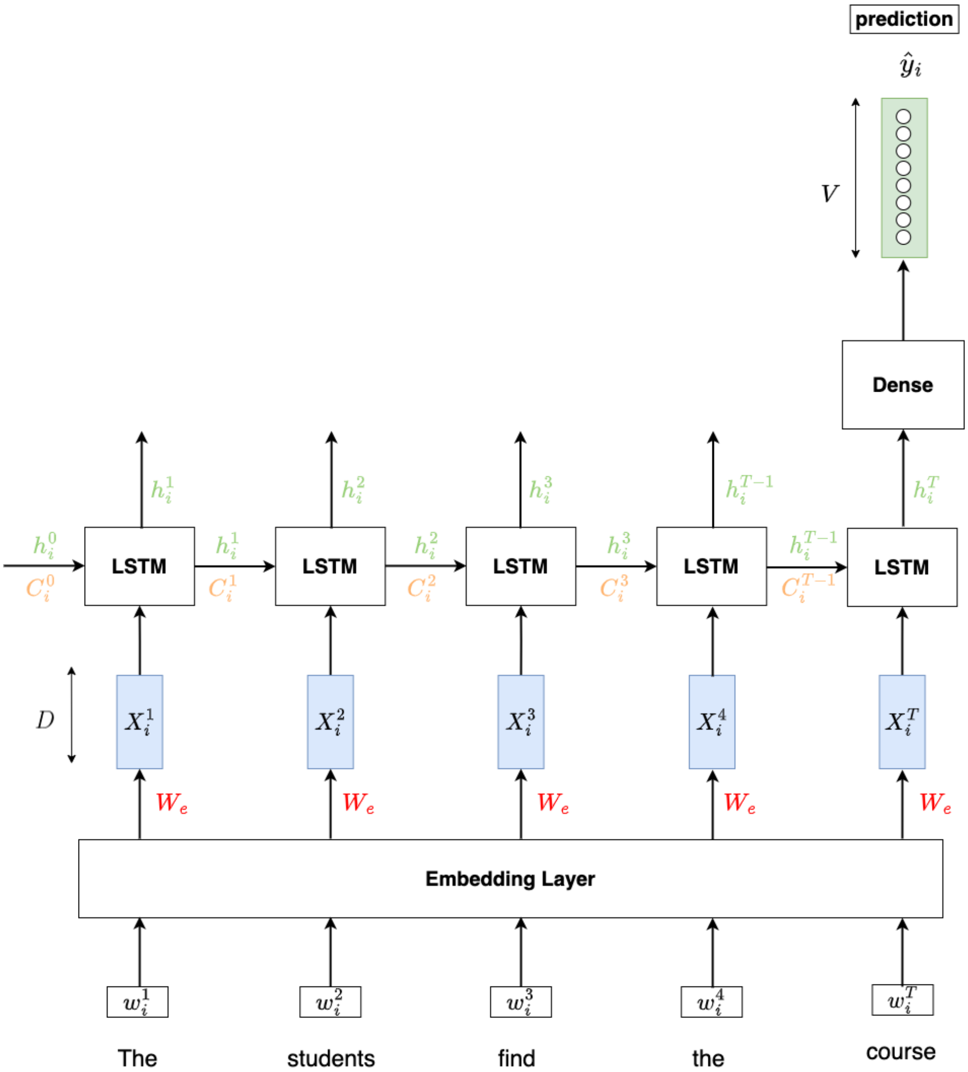

Figure 2: Predicting the next word

Let V be the vocabulary size. We would like to map the sequence of tokens $(w_i^1, \dots, w_i^T)$ associated with the sentence "The students find the course" to the next word:

- We use an embedding layer, parameterized by the matrix W_e, which gives the sequence of embedding vectors $(X_i^1, \dots, X_i^T)$ of dimension D.
- We use an LSTM layer with hidden states $(h_i^t, C_i^t)_{1 \leq t \leq T}$ of size M.
- The last hidden state h_i^T is then mapped to the prediction vector $\hat{y}_i \in \mathbb{R}^V$ using a Dense layer parameterized by (W_d, b_d).
- The prediction vector $\hat{y}_i$ is then compared to the true prediction $\tilde{y}_i \in \{0, 1\}^V$

Q12: List all the parameters of the architecture *[6 marks]*

Q13: Choose reasonable values of V, D, M. What would be the total number of trainable parameters if we let the model learn the embedding matrix ? *[4 marks]*

Q14: With the same hyperparameters, what would be the total number of trainable parameters if we use pre-trained embedding vectors *[4 marks]*

2.1.3 Limitations of the context-free embedding models

Consider the following two sentences:

- Sentence A: "**Python** is a famous programming language"

- Sentence B: "**Python** is one of the largest snake species"

Q15: Based on the embedding of the word "Python" in both sentences, explain why using a context-free embedding model such as Word2vec or GloVe is suboptimal to represent the meaning of the word "Python". *[4 marks]*

2.2 The Scaled Dot Product Attention Layer

We would like to create context-based representations of the embedding vectors $(X^1, \ldots, X^T)$ using a self attention layer, as shown in figure 3

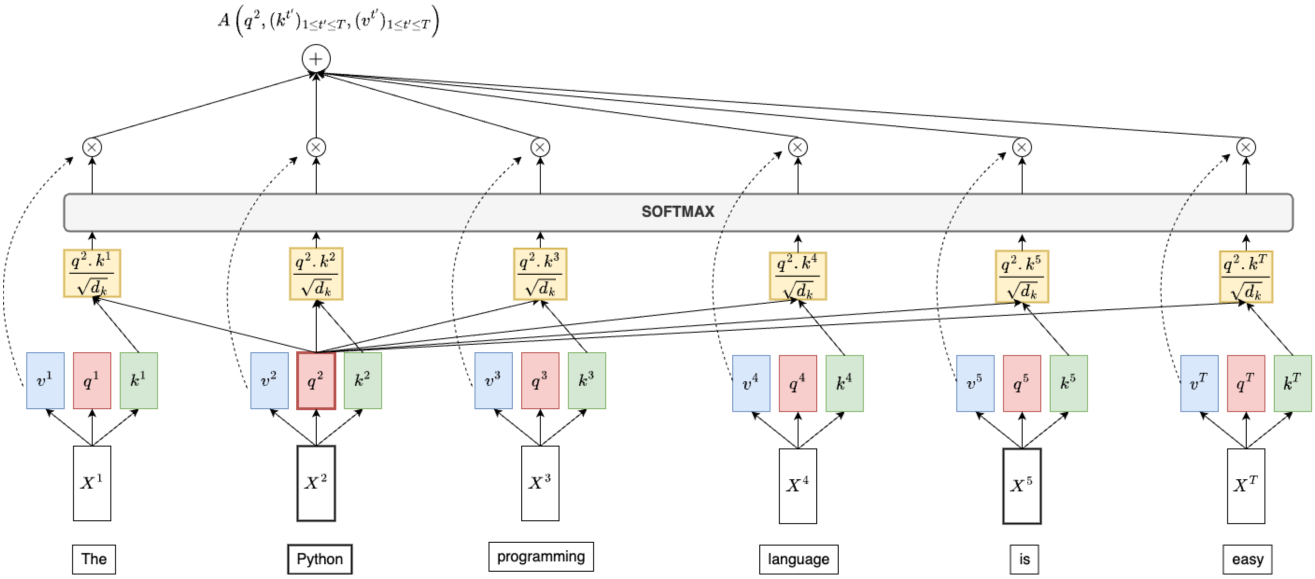

Figure 3: The Self Attention Layer

For all $t \in \{1, \ldots, T\}$, we define the projections of the embeddings X^t onto the d_q-dimensional query space, d_k-dimensional key space and d_v-dimensional value space as follows:

$$
\begin{aligned}
\mathbb{R}^{d_q} &\ni q^t = W_Q^T X^t \\
\mathbb{R}^{d_k} &\ni k^t = W_K^T X^t \\
\mathbb{R}^{d_v} &\ni v^t = W_V^T X^t
\end{aligned}
$$

Where $W_Q \in \mathbb{R}^{D\times d_q}$, $W_K \in \mathbb{R}^{D\times d_k}$ and $W_V \in \mathbb{R}^{D\times d_v}$ are the projection matrices onto the low dimensional query, key and value spaces, respectively. We also need $d_q = d_k$.

Let $A\left(q^2, (k^{t'})_{1\leq t'\leq T}, (v^{t'})_{1\leq t'\leq T}\right)$ be the context-based representation of the embedding vector X^2.

Q16: What is the expression of $A\left(q^2, (k^{t'})_{1\leq t'\leq T}, (v^{t'})_{1\leq t'\leq T}\right)$? *[4 marks]*

Q17: Does the context-based representation $A\left(q^2, (k^{t'})_{1\leq t'\leq T}, (v^{t'})_{1\leq t'\leq T}\right)$ depend on the order of the tokens in the sentence "The Python programming language is easy ? *[2 marks]*

We can generalize the way to create the context-based embedding $A\left(q^2, (k^{t'})_{1\leq t'\leq T}, (v^{t'})_{1\leq t'\leq T}\right)$ associated with the embedding X^2 to all the embedding vectors $(X^{t'})_{1\leq t'\leq T}$.

We consider the following matrices:

$$
Q = \begin{bmatrix} - & q^1 & - \\ \vdots & \vdots & \vdots \\ - & q^T & - \end{bmatrix} \in \mathbb{R}^{T\times d_q}, \quad K = \begin{bmatrix} - & k^1 & - \\ \vdots & \vdots & \vdots \\ - & k^T & - \end{bmatrix} \in \mathbb{R}^{T\times d_k}, \quad V = \begin{bmatrix} - & v^1 & - \\ \vdots & \vdots & \vdots \\ - & v^T & - \end{bmatrix} \in \mathbb{R}^{T\times d_v}
$$

We define the scaled dot product attention matrix, denoted $A(Q, K, V)$, as follows:

$$A(Q, K, V) := \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

Where the notation Softmax(M) for a matrix $M \in \mathbb{R}^{T \times d}$ refers to the Softmax applied to each row of the matrix M.

Q18: Show that: *[5 marks]*

$$A(Q, K, V) = \begin{bmatrix} - & A\left(q^1, (k^{t'})_{1 \leq t' \leq T}, (v^{t'})_{1 \leq t' \leq T}\right) & - \\ \vdots & \vdots & \vdots \\ - & A\left(q^t, (k^{t'})_{1 \leq t' \leq T}, (v^{t'})_{1 \leq t' \leq T}\right) & - \\ \vdots & \vdots & \vdots \\ - & A\left(q^T, (k^{t'})_{1 \leq t' \leq T}, (v^{t'})_{1 \leq t' \leq T}\right) & - \end{bmatrix}$$

In other words, the t-th row of the scaled dot product attention matrix $A(Q, K, V)$ is the context-based embedding vector $A\left(q^t, (k^{t'})_{1 \leq t' \leq T}, (v^{t'})_{1 \leq t' \leq T}\right)$ associated with the embedding vector X^t.

3 A Sequential Neural Network [25 marks]

In this section, we are dealing with a long sequence $X_1, X_2, \ldots, X_T$ of T continuous observations in $\mathbb{R}$. We wish to create a sequential neural network to predict the next observation based on the previous τ observations. Obviously $\tau < T$

- **Q19: How would you derive the training and validation data (both features and targets) from the long sequence $X_1, \ldots, X_T$?** *[5 marks]*
- **Q20: What are the main difficulties when training Recurrent Neural Networks? How can you handle them?***[5 marks]*
- **Q21: Describe the model you would use by specifying how the shape of the data is changing after each layer transformation.** *[7 marks]*
- **Q22: What would be your loss function?** *[3 marks]*
- **Q23: Describe the algorithm you would use for training your neural network.** *[5 marks]*