IAA: Deep Learning and NLP
: Exercises

IAA: Deep Learning and NLP : Exercises

To avoid overfitting, you can...

- ❑ increase the size of the network.
- ❑ use data augmentation.
- ❑ use Xavier initialization.
- ❑ stop training earlier.

IAA: Deep Learning and NLP : Exercises

To avoid overfitting, you can...

- ❑ increase the size of the network.
- ❑ use data augmentation.
- ❑ use Xavier initialization.
- ❑ stop training earlier.

To avoid overfitting, you can...

- ❑ increase the size of the network.
- ❑ use data augmentation.
- ❑ use Xavier initialization.
- ❑ stop training earlier.

IAA: Deep Learning and NLP : Exercises

Which of the following activation functions can lead to vanishing gradients?

(i) ReLU

(ii) Tanh

(iii) Leaky ReLU

(iv) None of the above

IAA: Deep Learning and NLP : Exercises

Which of the following activation functions can lead to vanishing gradients?
(i) ReLU
(ii) Tanh
(iii) Leaky ReLU
(iv) None of the above

Which of the following activation functions can lead to vanishing gradients?
(i) ReLU
(ii) Tanh
(iii) Leaky ReLU
(iv) None of the above

IAA: Deep Learning and NLP : Exercises

What is true about Dropout?

- ❑ The training process is faster and more stable to initialization when using Dropout.
- ❑ You should not use weaky ReLu as non-linearity when using Dropout.
- ❑ Dropout acts as regularization.
- ❑ Dropout is applied differently during training and testing.

IAA: Deep Learning and NLP : Exercises

What is true about Dropout?

- ❑ The training process is faster and more stable to initialization when using Dropout.
- ❑ You should not use weaky ReLu as non-linearity when using Dropout.
- ❑ Dropout acts as regularization.
- ❑ Dropout is applied differently during training and testing.

What is true about Dropout?

- ❑ The training process is faster and more stable to initialization when using Dropout.
- ❑ You should not use weaky ReLu as non-linearity when using Dropout.
- ❑ Dropout acts as regularization.
- ❑ Dropout is applied differently during training and testing.

IAA: Deep Learning and NLP : Exercises

What is true about Batch Normalization?

- ✓ Batch Normalization uses two trainable parameters that allow the network to undo the normalization eect of this layer if needed.
- ✓ Batch Normalization makes the gradients more stable so that we can train deeper networks.
- ✓ At test time, Batch Normalization uses a mean and variance computed on training samples to normalize the data.
- ✓ Batch Normalization has learnable parameters.

IAA: Deep Learning and NLP : Exercises

What is true about Batch Normalization?

- ✓ Batch Normalization uses two trainable parameters that allow the network to undo the normalization eect of this layer if needed.
- ✓ Batch Normalization makes the gradients more stable so that we can train deeper networks.
- ✓ At test time, Batch Normalization uses a mean and variance computed on training samples to normalize the data.
- ✓ Batch Normalization has learnable parameters.

What is true about Batch Normalization?

- ✓ Batch Normalization uses two trainable parameters that allow the network to undo the normalization eect of this layer if needed.
- ✓ Batch Normalization makes the gradients more stable so that we can train deeper networks.
- ✓ At test time, Batch Normalization uses a mean and variance computed on training samples to normalize the data.
- ✓ Batch Normalization has learnable parameters.

IAA: Deep Learning and NLP : Exercises

Which of the following optimization methods use first order momentum?

- ❑ Stochastic Gradient Descent
- ❑ Adam
- ❑ RMSProp
- ❑ Gauss-Newton

Which of the following optimization methods use first order momentum?

- ❑ Stochastic Gradient Descent
- ❑ Adam
- ❑ RMSProp
- ❑ Gauss-Newton

IAA: Deep Learning and NLP : Exercises

Making your network deeper by adding more parametrized layers will always...

- ✓ slow down training and inference speed.
- ✓ reduce the training loss.
- ✓ improve the performance on unseen data.

Making your network deeper by adding more parametrized layers will always...

- ✓ slow down training and inference speed.
- ✓ reduce the training loss.
- ✓ improve the performance on unseen data.

IAA: Deep Learning and NLP : Exercises

Explain why we need activation functions?

Without non-linearities, our network can only learn linear functions, because the composition of linear functions is again linear.

IAA: Deep Learning and NLP : Exercises

Which of the following is true about Batchnorm?

(i) Batchnorm is another way of performing dropout.

(ii) Batchnorm makes training faster.

(iii) In Batchnorm, the mean is computed over the features.

(iv) Batchnorm is a non-linear transformation to center the dataset around the origin

(i) Batchnorm is another way of performing dropout.

(ii) Batchnorm makes training faster.

(iii) In Batchnorm, the mean is computed over the features.

(iv) Batchnorm is a non-linear transformation to center the dataset around the origin

IAA: Deep Learning and NLP : Exercises

When should multi-task learning be used?

(i) When your problem involves more than one class label.

(ii) When two tasks have the same dataset.

(iii) When you have a small amount of data for a particular task that would benet from the large dataset of another task.

(iv) When the tasks have datasets of different formats (text and images).

IAA: Deep Learning and NLP : Exercises

When should multi-task learning be used?

(i) When your problem involves more than one class label.

(ii) When two tasks have the same dataset.

(iii) When you have a small amount of data for a particular task that would benet from the large dataset of another task.

(iv) When the tasks have datasets of different formats (text and images).

(i) When your problem involves more than one class label.

(ii) When two tasks have the same dataset.

(iii) When you have a small amount of data for a particular task that would benet from the large dataset of another tas

(iv) When the tasks have datasets of different formats (text and images).

IAA: Deep Learning and NLP : Exercises

What is Error Analysis?

The process of analyzing the performance of a model through metrics such as precision, recall or F1-score.

(ii) The process of scanning mis-classied examples to identify weaknesses of a model.

(iii) The process of tuning hyperparameters to reduce the loss function during training.

(iv) The process of identifying which parts of your model contributed to the error.

IAA: Deep Learning and NLP : Exercises

What is Error Analysis?

The process of analyzing the performance of a model through metrics such as precision, recall or F1-score.

(ii) The process of scanning mis-classied examples to identify weaknesses of a model.

(iii) The process of tuning hyperparameters to reduce the loss function during training.

(iv) The process of identifying which parts of your model contributed to the error.

The process of analyzing the performance of a model through metrics such as precision, recall or F1-score.

(ii) The process of scanning mis-classied examples to identify weaknesses of a model.

(iii) The process of tuning hyperparameters to reduce the loss function during training.

(iv) The process of identifying which parts of your model contributed to the error.

IAA: Deep Learning and NLP : Exercises

After training a neural network, you observe a large gap between the training accuracy (100%) and the test accuracy (42%). Which of the following methods is commonly used to reduce this gap?

(i) Generative Adversarial Networks

(ii) Dropout

(iii) Sigmoid activation

(iv) RMSprop optimizer

IAA: Deep Learning and NLP : Exercises

After training a neural network, you observe a large gap between the training accuracy (100%) and the test accuracy (42%). Which of the following methods is commonly used to reduce this gap?

(i) Generative Adversarial Networks

(ii) Dropout

(iii) Sigmoid activation

(iv) RMSprop optimizer

(i) Generative Adversarial Networks

(ii) Dropout

(iii) Sigmoid activation

(iv) RMSprop optimizer

IAA: Deep Learning and NLP : Exercises

In DL architecture, we usually implement a `forward` and a `backward` function for each layer.

Explain what the "**Forward**" function do,

- potential variables that they need to save, which arguments they take, and what they return.?

- ***takes output from previous layer, performs operation, returns result***
- ***caches values needed for gradient computation during backprop.***

IAA: Deep Learning and NLP : Exercises

In DL architecture, we usually implement a `forward` and a `backward` function for each layer.

Explain what the "**backward**" function do,

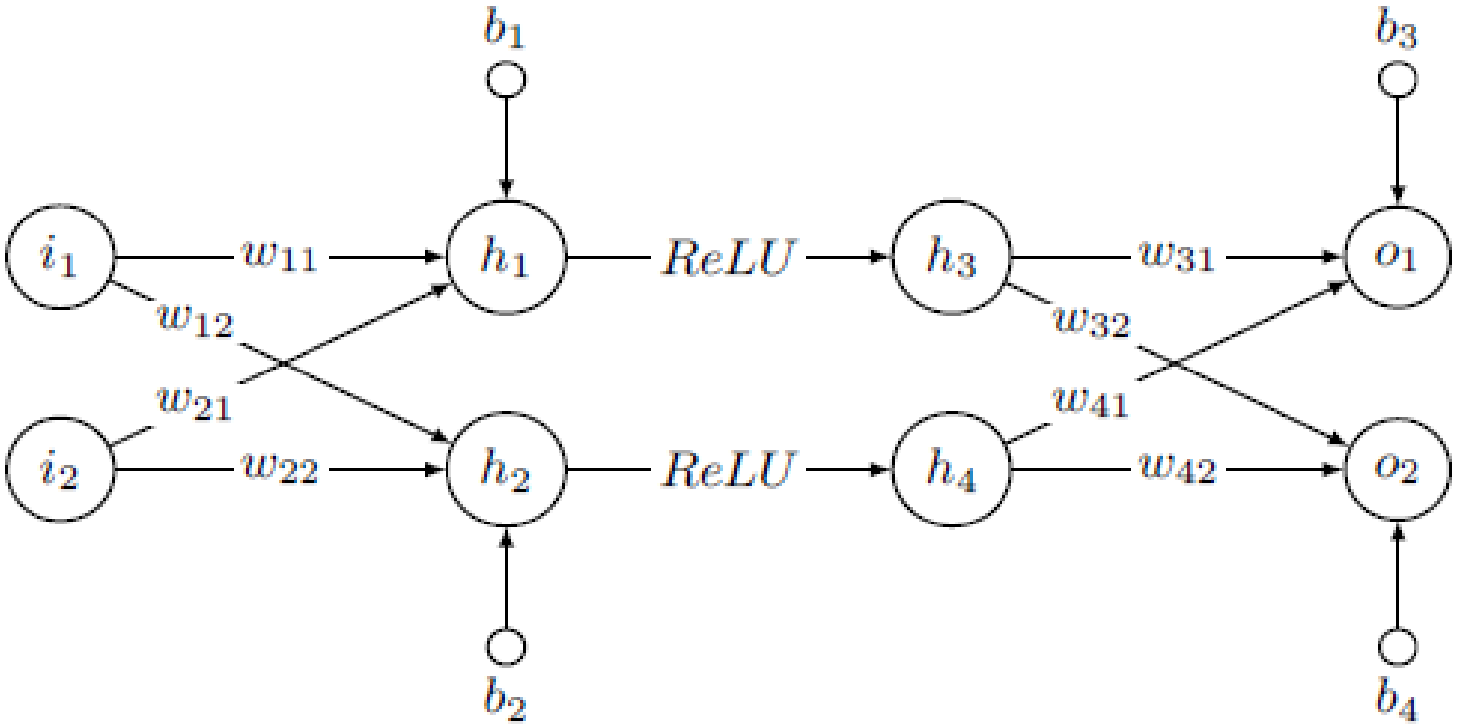
- potential variables that they need to save, which arguments they take, and what they return.?

- ***takes upstream gradient, returns all partial derivatives.***

IAA: Deep Learning and NLP : Exercises

Given the following neural network with fully connection layer and ReLU activations, including two input units (i1; i2), four hidden units (h1; h2) and (h3; h4). The output units are indicated as (o1; o2) and their targets are indicated as (t1; t2). The weights and bias of fully connected layer are called **w** and b with specific sub-descriptors.

Compute the output (o1; o2) with the input (i1; i2) and network parameters as specified above. Write down all calculations, including intermediate layer results.



- ***The values of variables are given in the following***

i_1	i_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	w_{41}	w_{42}	b_1	b_2	b_3	b_4	t_1	t_2
2.0	-1.0	1.0	-0.5	0.5	-1.0	0.5	-1.0	-0.5	1.0	0.5	-0.5	-1.0	0.5	1.0	0.5

IAA: Deep Learning and NLP : Exercises

$$h_1 = i_1 \times w_{11} + i_2 \times w_{21} + b_1 = 2.0 \times 1.0 - 1.0 \times 0.5 + 0.5 = 2.0$$

$$h_2 = i_1 \times w_{12} + i_2 \times w_{22} + b_2 = 2.0 \times -0.5 + -1.0 \times -1.0 - 0.5 = -0.5$$

Forward pass:

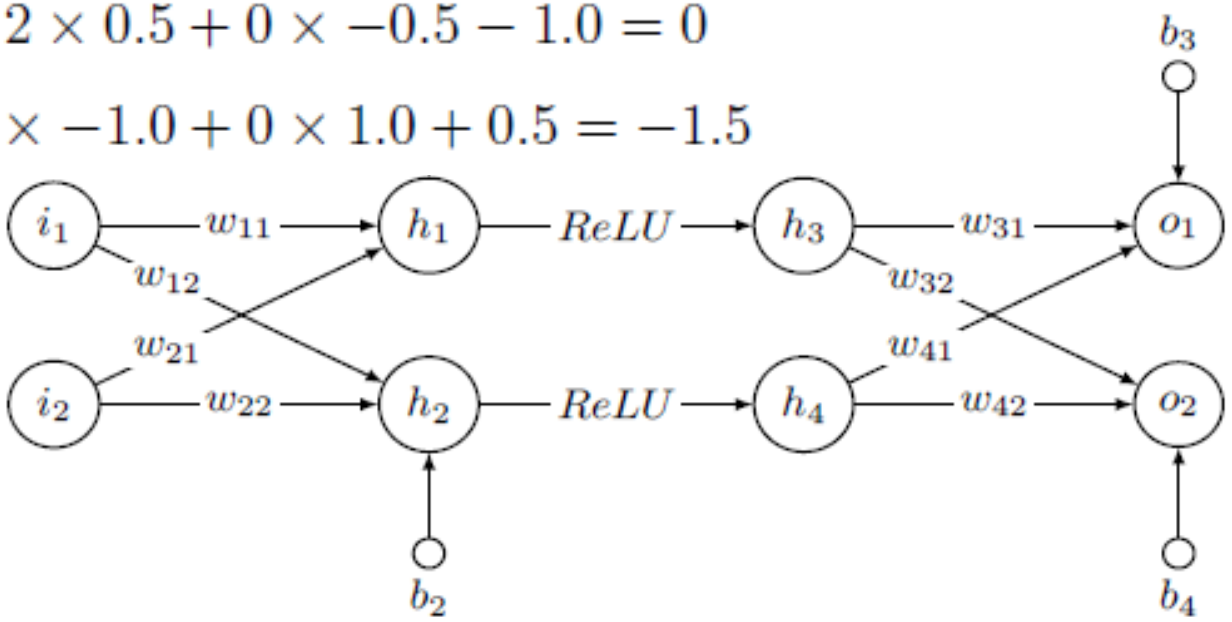
$$h_3 = max(0, h_1) = h_1 = 2$$

$$h_4 = max(0, h_2) = 0$$

$$o_1 = h_3 \times w_{31} + h_4 \times w_{41} + b_3 = 2 \times 0.5 + 0 \times -0.5 - 1.0 = 0$$

$$o_2 = h_3 \times w_{32} + h_4 \times w_{42} + b_4 = 2 \times -1.0 + 0 \times 1.0 + 0.5 = -1.5$$

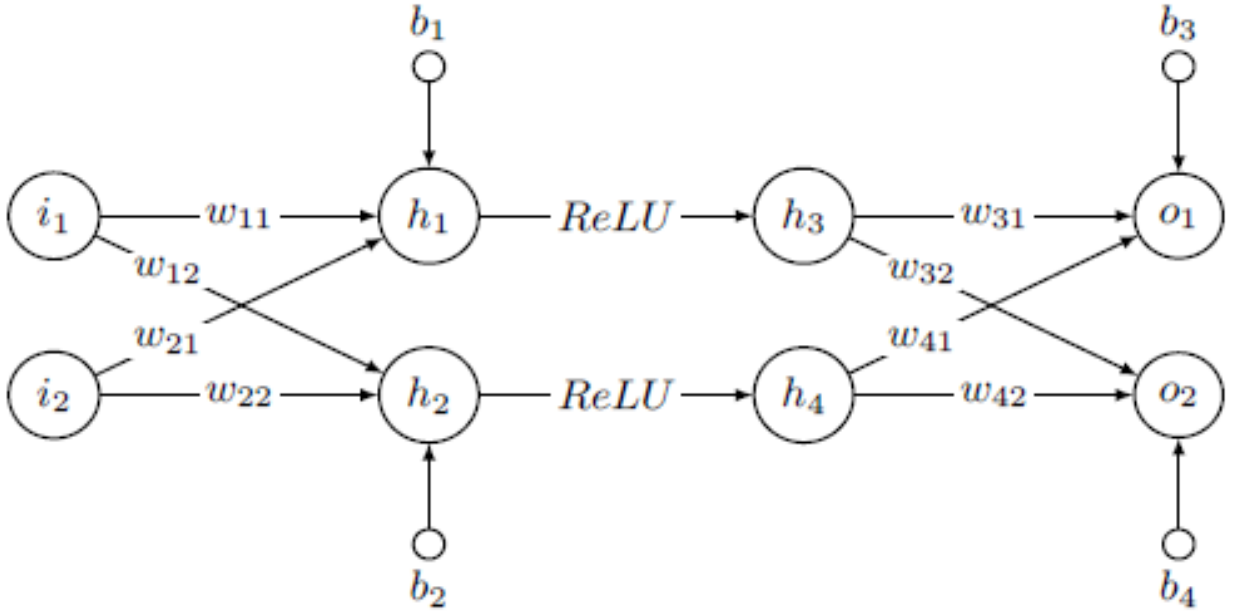
Compute the output (o1; o2) with the input (i1; i2) and network parameters as specified above. Write down all calculations, including intermediate layer results.



i_1	i_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	w_{41}	w_{42}	b_1	b_2	b_3	b_4	t_1	t_2
2.0	-1.0	1.0	-0.5	0.5	-1.0	0.5	-1.0	-0.5	1.0	0.5	-0.5	-1.0	0.5	1.0	0.5

IAA: Deep Learning and NLP : Exercises

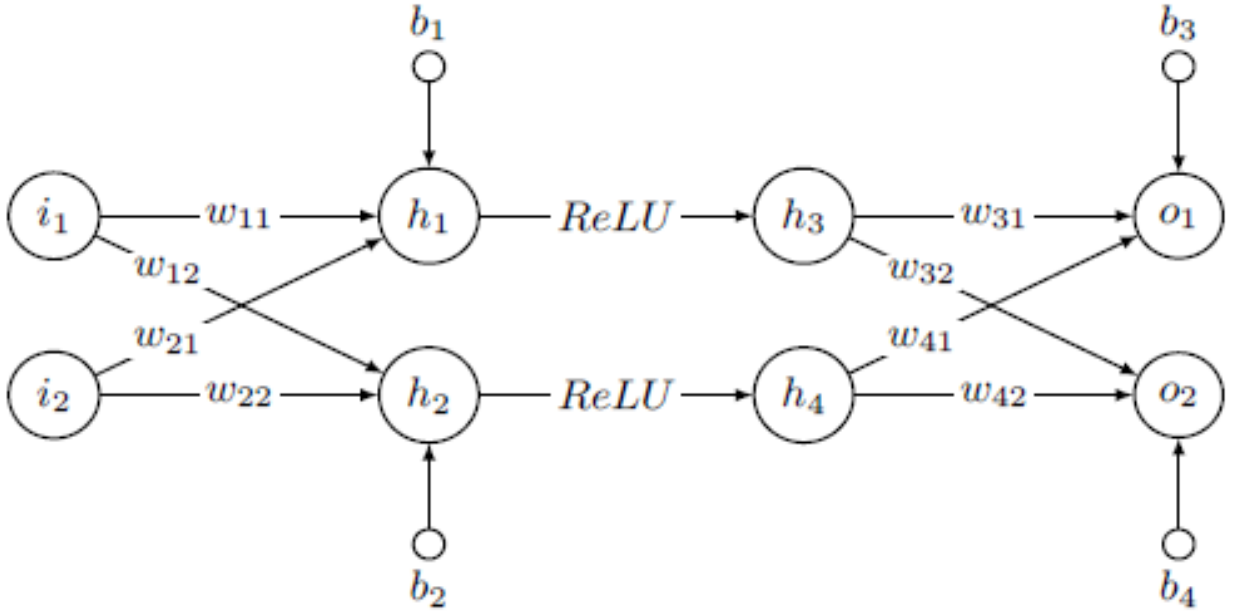
Compute the mean squared error of the output (o1; o2) calculated above and the target (t1; t2).



i_1	i_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	w_{41}	w_{42}	b_1	b_2	b_3	b_4	t_1	t_2
2.0	-1.0	1.0	-0.5	0.5	-1.0	0.5	-1.0	-0.5	1.0	0.5	-0.5	-1.0	0.5	1.0	0.5

IAA: Deep Learning and NLP : Exercises

Compute the mean squared error of the output (o1; o2) calculated above and the target (t1; t2).



$$MSE = \frac{1}{2} \times (t_1 - o_1)^2 + \frac{1}{2} \times (t_2 - o_2)^2 = 0.5 \times 1.0 + 0.5 \times 4.0 = 2.5$$

i_1	i_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	w_{41}	w_{42}	b_1	b_2	b_3	b_4	t_1	t_2
2.0	-1.0	1.0	-0.5	0.5	-1.0	0.5	-1.0	-0.5	1.0	0.5	-0.5	-1.0	0.5	1.0	0.5

IAA: Deep Learning and NLP : Exercises

- Give one advantage of **GloVe** over **Skipgram/CBOW** models.

- What are two ways practitioners deal with having two different sets of word vectors U and V at the end of training both Glove and word2vec?

  Fr: Quelles sont les deux façons dont les praticiens gèrent le fait d'avoir deux ensembles différents de vecteurs de mots U et V à la fin de l'entraînement des modèles GloVe et Word2Vec ?

- ……suffer(s) from the vanishing gradient problem.
  Circle all that apply and JUSTIFY YOUR ANSWER.

1. 1-Layer Feed-forward NN (i.e., the NN from problem set 1)
2. Very Deep Feed-forward NN
3. Recurrent NN
4. Word2vec CBOW
5. Word2vec Skip-Gram

IAA: Deep Learning and NLP : Exercises

- Give one advantage of **GloVe** over **Skipgram/CBOW** models.
  Fast training

- What are two ways practitioners deal with having two different sets of word vectors U and V at the end of training both Glove and word2vec?

  Fr: Quelles sont les deux façons dont les praticiens gèrent le fait d'avoir deux ensembles différents de vecteurs de mots U et V à la fin de l'entraînement des modèles GloVe et Word2Vec ?

  1. Add them
  2. Concatenate them.

- ……suffer(s) from the vanishing gradient problem.
  Circle all that apply and JUSTIFY YOUR ANSWER.

1. 1-Layer Feed-forward NN (i.e., the NN from problem set 1)
2. Very Deep Feed-forward NN
3. Recurrent NN
4. Word2vec CBOW
5. Word2vec Skip-Gram

Very Deep Feed-forward NN, Recurrent NN. All of these models are "deep", involving chained matrix multiplication and possibly saturating non-linearities.

IAA: Deep Learning and NLP : Exercises

- If your training loss increases with number of epochs, which of the following could be a possible issue with the learning process?

- Regularization is too low and model is overfitting
- Regularization is too high and model is underfitting
- Step size is too large
- Step size is too small

IAA: Deep Learning and NLP : Exercises

- If your training loss increases with number of epochs, which of the following could be a possible issue with the learning process?

- Regularization is too low and model is overfitting
- Regularization is too high and model is underfitting
- Step size is too large
- Step size is too small

The train loss always decreases whether the model is overfitting or underfitting. If the step size is too small, the convergence is too slow, but the training loss will still go down. If the step size is too large, it may cause a bouncing effect because we skip and overshoot the optimal solution. This leads to increase in training loss and decrease in training accuracy.

IAA: Deep Learning and NLP : Exercises

Dense word vectors learned through **word2vec** or **GloVe** have many advantages over using sparse **one-hot word** vectors. Which of the following is NOT an advantage dense vectors have over sparse vectors?

A. Models using **dense word vectors** generalize better to unseen words than those using **sparse vectors**.
B. Models using **dense word vectors** generalize better to rare words than those using **sparse vectors**.
C. **Dense word vectors** encode similarity between words while **sparse vectors** do not.
D. **Dense word vectors** are easier to include as features in machine learning systems than **sparse vectors**.

IAA: Deep Learning and NLP : Exercises

Dense word vectors learned through **word2vec** or **GloVe** have many advantages over using sparse **one-hot word** vectors. Which of the following is NOT an advantage dense vectors have over sparse vectors?

A. Models using **dense word vectors** generalize better to unseen words than those using **sparse vectors**.
B. Models using **dense word vectors** generalize better to rare words than those using **sparse vectors**.
C. **Dense word vectors** encode similarity between words while **sparse vectors** do not.
D. **Dense word vectors** are easier to include as features in machine learning systems than **sparse vectors**.

Solution: A. Just like sparse representations, word2vec or GloVe do not have representations for unseen words and hence do not help in generalization.

IAA: Deep Learning and NLP : Exercises

Which of the following statements is INCORRECT?

A. Recurrent neural networks can handle a sequence of arbitrary length, while feedforward neural networks can not.
B. Training recurrent neural networks is hard because of vanishing and exploding gradient problems.
C. Gradient clipping is an eective way of solving vanishing gradient problem.
D. Gated recurrent units (GRUs) have fewer parameters than LSTMs.

IAA: Deep Learning and NLP : Exercises

Which of the following statements is INCORRECT?

A. Recurrent neural networks can handle a sequence of arbitrary length, while feedforward neural networks can not.
B. Training recurrent neural networks is hard because of vanishing and exploding gradient problems.
C. Gradient clipping is an eective way of solving vanishing gradient problem.
D. Gated recurrent units (GRUs) have fewer parameters than LSTMs.

Solution: **C. Gradient clipping is only a solution for solving exploding gradient problems, not vanishing gradient ones.**

IAA: Deep Learning and NLP : Exercises

True OR False

- ✓ A multi-layer neural network model trained using stochastic gradient descent on the same dataset with different initializations for its parameters is guaranteed to learn the same parameters.

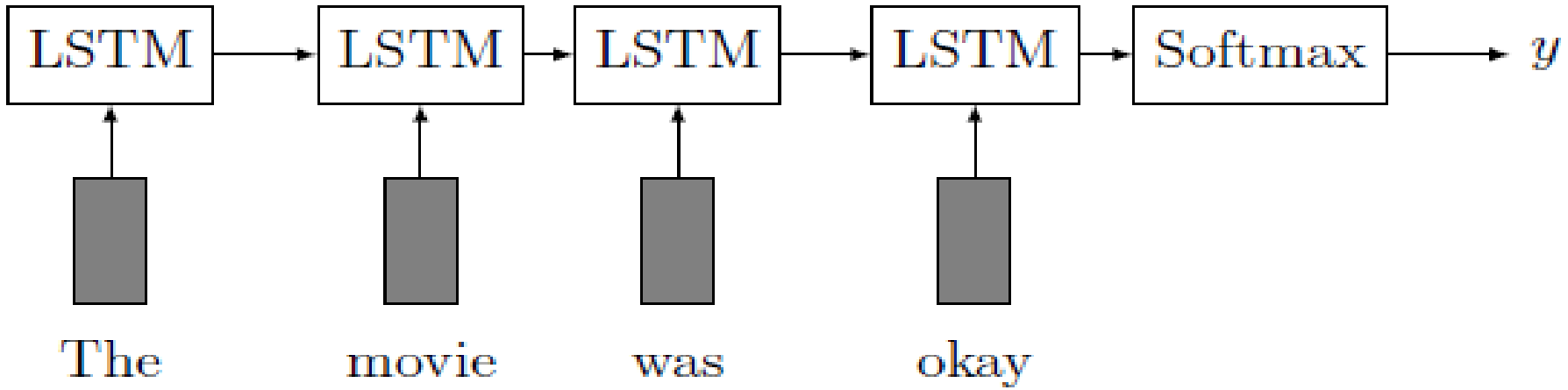
Solution: False. The loss function for a multi-layer neural network is non-convex and hence SGD is only guaranteed to converge to a local optimum.

- ✓ Stochastic gradient descent results in a smoother convergence plot (loss vs epochs) as compared to batch gradient descent.

Solution: False. SGD results in noisier convergence plots compared to batch gradient descent.

IAA: Deep Learning and NLP : Exercises

A popular model used for sentiment classification is an LSTM model:

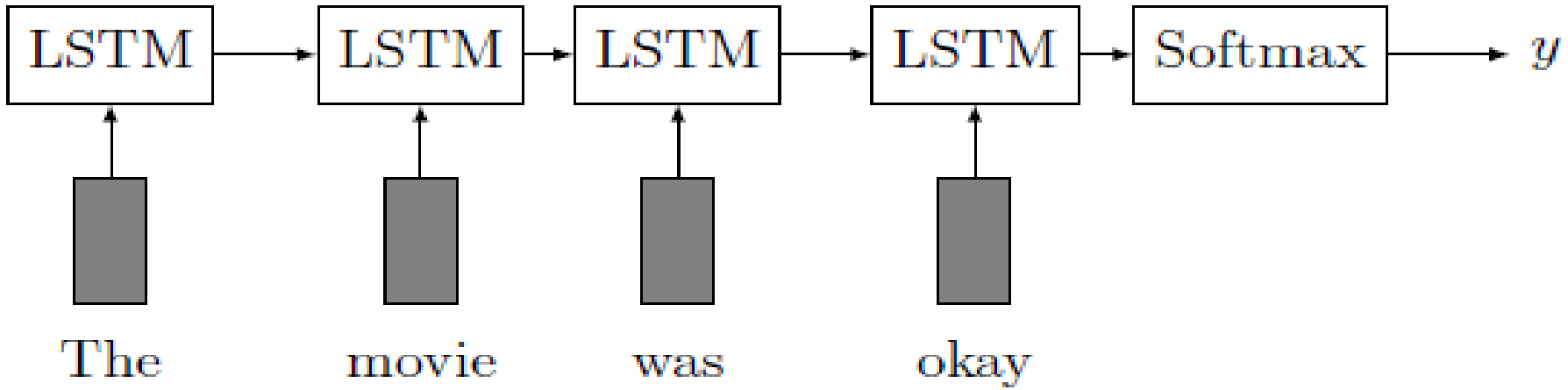


This model inputs word vectors to the LSTM model at each time step and uses the last hidden state vector to predict the sentiment label (y).
Recall that in "bag-of-vectors" model for sentiment classification, we use the average of all the word vectors in a sentence to predict the sentiment label.

- Name at least one benefit of the LSTM model over the bag-of-vectors model.
- If we chose to update our word vectors when training the LSTM model on sentiment classification data, how would these word vectors differ from ones not updated during training? Explain with an example. Assume that the word vectors of the LSTM model were initialized using GloVe or word2vec.

IAA: Deep Learning and NLP : Exercises

A popular model used for sentiment classification is an LSTM model:



- ✓ The LSTM model is able to integrate information from word ordering, e.g. "*this was not an amazing fantastic movie*" , while the bag-of-vectors model can not.

- ✓ The word vectors learned using this method can capture more sentiment information. In the word2vec models that depend on co-occurrence counts, words like `good' or `bad' have very similar representations; representations learned through a sentiment classier would learn different embeddings.

Transformer Architecture

- A Transformer architecture, taken from "Attention is All You Need", is shown to the right
- The encoder and decoder both rely on *stacked layers*
- Each layer has sublayers
- Some of the sublayers rely on a concept called **self-attention** (I will briefly discuss this in class)
- Others are point-wise (a.k.a. position-wise), fully-connected, feedforward NNs
- There are also skip connections around the sublayers
- Note that positional encodings are added to the input and output token embeddings
- This allows transformers to be dependent on word order

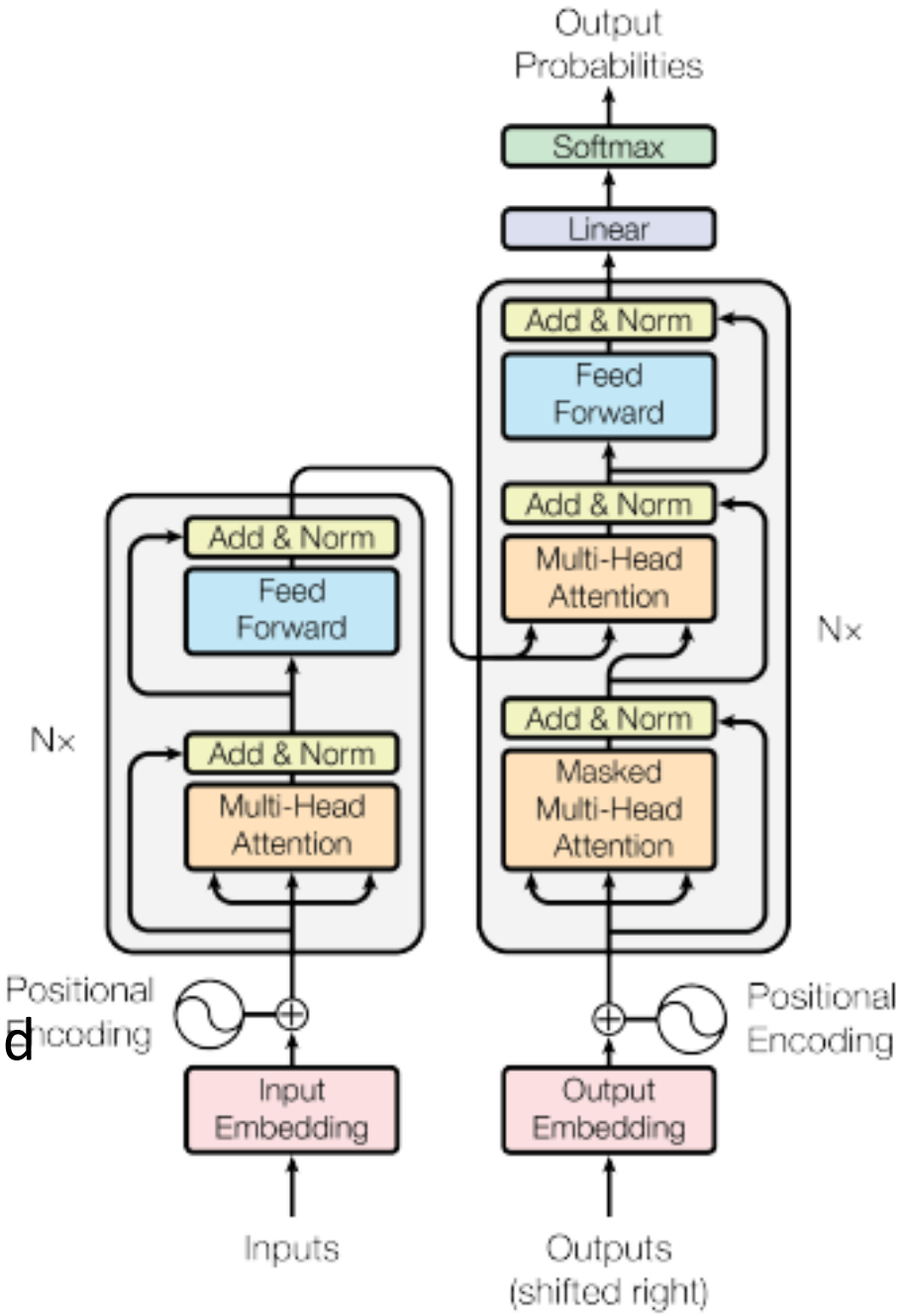


Figure 1: The Transformer - model architecture.

IAA: Deep Learning and NLP : Exercises

- ✓ What was the primary purpose of introducing the attention mechanism?

A. To reduce the training time of models
B. To improve the model's ability to process long sequences
C. To increase the number of layers in the model
D. To implement the model's auto-encoding functionality

- ✓ Explain the application of attention mechanisms in sequence models and discuss how they help enhance model performance and advantages.

IAA: Deep Learning and NLP : Exercises

✓ What was the primary purpose of introducing the attention mechanism?

A. To reduce the training time of models
B. To improve the model's ability to process long sequences
C. To increase the number of layers in the model
D. To implement the model's autoencoding functionality

The attention mechanism was primarily introduced to enhance the model's ability to handle long sequences by enabling it to focus on the key information within the sequence, thereby improving its performance.

✓ Explain the application of attention mechanisms in sequence models and discuss how they help enhance model performance and advantages.

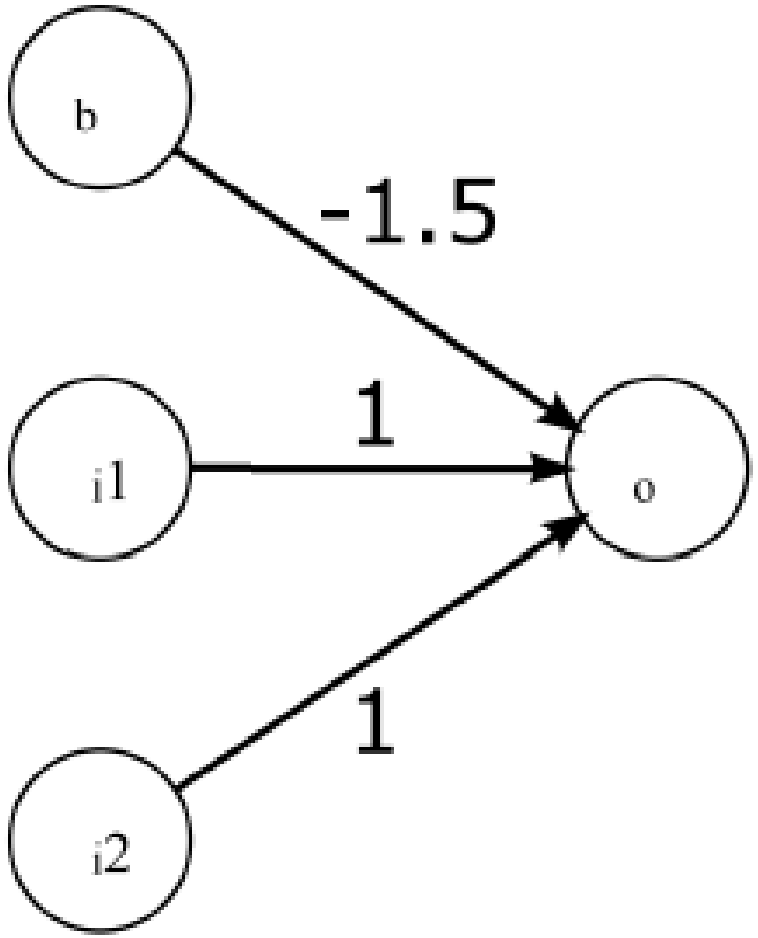
In sequence models, the attention mechanism allows the model to concentrate on the most important parts of the input sequence, thereby enhancing performance. For instance, in machine translation, the model can focus on the source language words most relevant to the word being translated at any given moment, thereby improving the accuracy of translation and the ability to handle long-distance dependencies. The advantages of this mechanism include improved model interpretability and flexibility, as well as increased efficiency and accuracy when dealing with complex or lengthy sequence data.

•IAA: Deep Learning and NLP : Exercises

In the perceptron below, what will the output be when the input is **(0, 0)?**

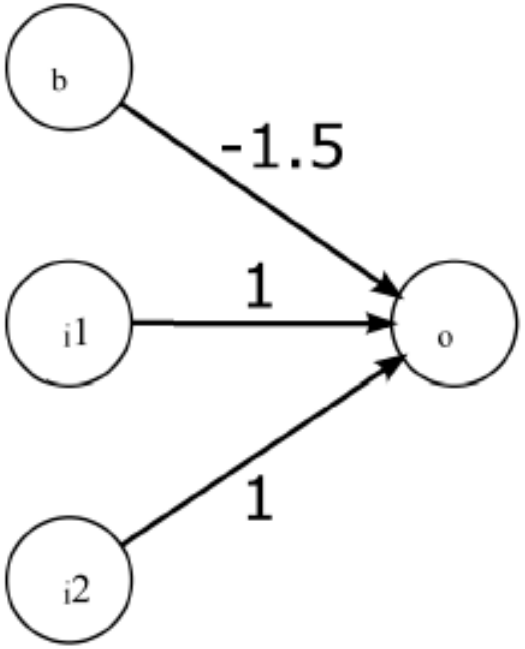
What about inputs **(0, 1), (1, 1)** and **(1, 0)?**

What if we change the bias weight to **-0.5**?



•IAA: Deep Learning and NLP : Exercises

In the perceptron below, what will the output be when the input is **(0, 0)**
What about inputs **(0, 1), (1, 1)** and **(1, 0)?**
What if we change the bias weight to **-0.5**?



Bias $= -1.5$			Bias $= -0.5$		
Input	Weighted sum	Output	Input	Weighted sum	Output
$(0,0)$	-1.5	0	$(0,0)$	-0.5	0
$(0,1)$	-0.5	0	$(0,1)$	0.5	1
$(1,0)$	-0.5	0	$(1,0)$	0.5	1
$(1,1)$	0.5	1	$(1,1)$	1.5	1

•IAA: Deep Learning and NLP : Exercises

Multi Layer Perceptron (MLP)

Suppose we have a multilayer perceptron (MLP) model with 17 neurons in the input layer, 25 neurons in the hidden layer and 10 neuron in the output layer. What is the size of the weight matrix between the hidden layer and output layer?

(a) 25 x 17
(b) 10 x 25
(c) 25 x 10
(d) 17 x 10

•IAA: Deep Learning and NLP : Exercises

What does the PyTorch optimizer's step() function do when training neural networks?

(a) Adjust the network's weights based on the gradients
(b) Randomly initializing the network's weights.
(c) Sets all the network's gradients to zero to prepare it for backpropagation
(d) Compute the gradients of the network based on the error between predicted and actual outputs.

IAA: Deep Learning and NLP : Exercises

Consider the following neural network with weights shown in the image below. Every hidden neuron uses the **ReLU** activation function, and there is no activation function on the output neuron.
Assume there are **no bias terms**. What is the output of this network with the input **x = (1; 2)?**

Give a numerical answer.

The answer is -3.