EN4554: End-Semester Exam

Circle the most accurate answer to each question (2 marks each).

1. Which statement is **true** about the correlation and dependence of two random variables?

   (a) If they are uncorrelated they must be independent.
   (b) If they are independent they must be uncorrelated.
   (c) Both of the above.
   (d) None of the above.

2. What is an assumption of the Ordinary Least Squares (OLS) regression?

   (a) Inputs are normally distributed.
   (b) Outputs are normally distributed.
   (c) Weights are normally distributed.
   (d) Prediction errors are normally distributed.

3. You are working on a linear regression problem with a dataset that has a large number of outliers. What is the best precaution you can take?

   (a) Normalize the input features by subtracting the mean and dividing by the standard deviation
   (b) Use L1 regularization on the weights
   (c) Use the mean-squared-error as the loss function
   (d) Use the Huber loss as the loss function

4. Which statement is **false** about the L1 and L2 weight regularizations of a neural network?

   (a) L1 regularization can be used with the cross-entropy loss.
   (b) L2 regularization cannot be used when dropout is used.
   (c) L1 regularization promotes weight sparsity.
   (d) L2 regularization promotes smaller weights.

5. Your image classification network has high train accuracy but low validation accuracy. Which of the following could fix the problem?

   (a) Use a larger neural network
   (b) Increase the dropout rate in the dense layers
   (c) Decrease the weight given to the L2 regularization loss
   (d) Train the network for longer

6. Which is **false** about the bias-variance decomposition of a neural network's generalization error.

(a) Increasing regularization strength can be a solution for high-bias.

(b) Using a bigger model can be a solution for high-bias.

(c) Using a bigger train set can be a solution for high-variance.

(d) Early stopping can be a solution for high-variance.

7. Which of the following is **false** about the sigmoid activation function used in a neural net?

   (a) It can cause vanishing gradient issues when used in the middle of a network.

   (b) It can cause exploding gradient issues when used in the middle of a network.

   (c) It can be used to convert an output neuron to a binary probability.

   (d) It can be used to introduce non-linearity to neural network.

8. What is **false** about data augmentation in image data?

   (a) It can reduce overfitting.

   (b) It is always better to randomize data augmentation operations.

   (c) It is not effective when image pixels are normalized to have zero mean, unit variance.

   (d) In some cases, performing data augmentation on CPU can cut down the overall training time.

We can do data augmentation on CPU while GPUs/TPUs are busy doing the NN computations.

9. What statement is **true** about the SGD with momentum optimizer?

   (a) It has only one hyperparameter: the learning rate. This is SGD

   (b) It effectively scales the learning rate to compensate for different gradient magnitudes in different directions. This is RMSProp

   (c) It smooths the gradients over time before applying the weight update rule.

   (d) It combines the ideas in Adam and RMSProp optimizers. Adam combines RMSProp & momentum

10. You have discovered that a particular CNN model trains well with batch size 32 and a flat learning rate of 1×10^{-3}. You now want to increase the batch size to 256 to finish the required number of epochs more quickly. What is the most appropriate adjustment you can make to the learning rate?

    (a) Keep it at 1×10^{-3} LR should increase with the batch size. But starting at a high lr could cause instability, hence the ramp up.

    (b) Use a new base learning rate 1.25×10^{-4}, while linearly ramping up to this value

    (c) Use a new base learning rate 8×10^{-3}, while linearly ramping up to this value

    (d) Switch between 1.25×10^{-4} and 8×10^{-3} after each epoch

11. A node in a neural network takes a real number x as input and outputs another real number y. The operation inside the node is simply squaring the input (therefore, $y = x^2$). During the forward pass with a single datum, x was equal to 0.6. During the corresponding backward pass, dE/dy (the error derivative w.r.t. y) is equal to 5.0. What is the value of dE/dx at this point?

    (a) 6.0

    (b) 2.24

    (c) 25.0

    (d) 0.36

$$\frac{dE}{dx} = \frac{dE}{dy} \times \frac{dy}{dx} = \frac{dE}{dy} 2x$$

$$= 5.0 \times 2 \times 0.6 = 6.0$$

12. In a semantic segmentation network, which of the following operations can significantly increase the field of view of a pixel in an intermediate feature map without losing spatial resolution?

    (a) Non-overlapping max pooling 3 Reduces resolution

    (b) Non-overlapping min pooling

(c) Dilated convolution

(d) Transposed convolution

13. Which statement is **false** about a CNN-based semantic segmentation networks?

    (a) They usually have and encoder part and a decoder part.

    (b) They can use skip connections to promote gradient flow at high resolutions.

    (c) They can use max-pooling to increase the field of view towards the beginning of the network.

    (d) They can use convolution with a stride greater than 1 to increase the resolution towards the end of the network. To increase res, we need fractional strides < 1 (Transposed Conv.)

14. Which of the following is **not** a straightforward use case (*i.e.*, a use case that does not require any additional deep networks) of an image-text model like CLIP, which learns to embed images and text into the same embedding space?

    (a) Generating new images

    (b) Zero-shot learning

    (c) Retrieve related images, given a text query — Store image embeddings, do a NN search with a query embedding.

    (d) Retrieve related texts, given an image query — the same thing the other way around.

15. You are training a ResNet-56 based image classification model in a fully-supervised setting to recognize different pet breeds. You have a limited amount of your own training data. What is a good way to obtain initial weights for the ResNet-56 backbone of your network, assuming all these are publicly available without licensing restrictions?

    (a) Take weights from a fully-supervised model trained on a large dataset such as ImageNet

    (b) Take weights from a self-supervised, contrastive learning method such as SimCLR

    (c) Take the image encoder of an image-text model such as CLIP

    (d) All of the above

Answer all questions. Please be concise.

1. (20 points) We discuss some basics about random variables and parameter estimation in this question.

   (a) (4 points) Let $Y \in (-\infty, \infty)$ be a continuous random variable with non-zero probability density throughout its support. Let Y_1 and Y_2 be two independent random variables with the same distribution as Y. You observe a special property that 2Y has the same distribution as $Y_1 + Y_2$. Prove that the variance of Y must be infinite.

   (b) Let $X \in \{1, 2, 3, ...\}$ be a discrete random variable with the probability mass function,

   $$\Pr(X = k) = \theta(1-\theta)^{k-1},$$

   where $\theta \in (0, 1]$ is a parameter. We are going to estimate θ based on realizations of X.

      i. (3 points) You have collected a dataset of n independent realizations of X given by $\{k_1, k_2, \ldots, k_n\}$. Write down the likelihood of your data in terms of $k_1, k_2, \ldots, k_n$ and θ.

      ii. (5 points) Using the result derived above, show that the maximum likelihood estimate (MLE) of θ is given by $\hat{\theta}_n = \frac{n}{\sum_{i=1}^{n} k_i}$ [**Hint**: Maximize the log-likelihood function w.r.t. θ.].

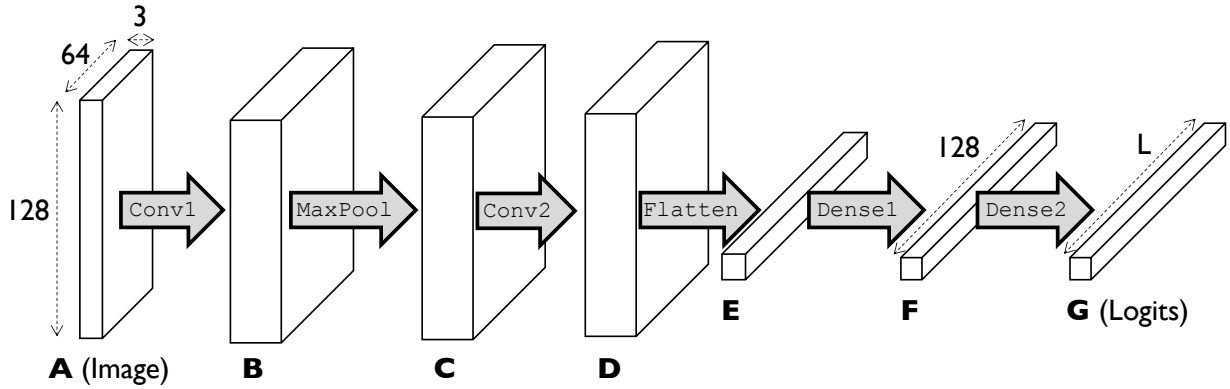
      iii. (5 points) Consider the case where $n = 1$. Your MLE estimator would then be $\hat{\theta}_1 = 1/k_1$, where k_1 comes from X's distribution. Starting from the definition of the expected value of a function of a random variable, show that,

      $$E[\hat{\theta}_1] = \theta + \frac{1}{2}\theta(1-\theta) + \frac{1}{3}\theta(1-\theta)^2 + \ldots .$$

iv. (3 points) What can you say about the bias of the estimator $\hat{\theta}_1$?

2. (20 points) You are working on a face recognition task for a client. You have a dataset of closely cropped RGB face images of size 128(height) × 64(width), belonging to 5,000 different people. For each image, you know the identity of the person. Your initial goal is to build a face classifier with fully-supervised training, that can classify a new image into one of the 5,000 classes (people) in your dataset.

   You use the CNN shown below to solve this problem. The tensors flowing through the network are named **A**, **B**, ..., **G**. We use the convention (`height`, `width`, `channels`) to denote tensor shapes and ignore the batch dimension. For example, the input image tensor **A** has shape (128, 64, 3). Tensor sizes in the diagram are not proportional to their actual shapes.



   The operations (layers) that transforms tensors are described below. Note that each layer has additive bias parameters whenever applicable.

   - `Conv1`: A 2D convolution operation with kernel size (3 × 3), stride 1 in both directions, 32 output channels, and 0-padding as necessary to compensate for any loss of spatial resolution at boundaries.
   - `MaxPool`: A 2D max-pool operation with kernel size (4 × 4), non-overlapping (that is, with stride 4 in both directions), without any 0-padding at boundaries.
   - `Conv2`: The same configuration as `Conv1`, but with 64 output channels.
   - `Flatten`: This flattens the 3-dimensional array input (tensor **D**) into a 1-dimensional array (tensor **E**).
   - `Dense1`: A dense (fully-connected) layer that outputs 128 units.
   - `Dense2`: A dense (fully-connected) layer that outputs L units.

   (a) (2 points) What should be the value of L?

   (b) (4 points) Write down the shapes of tensors **B** and **C** using the convention (`height`, `width`, `channels`).

   (c) (4 points) Write down the number of learnable parameters (parameters) in each of the following layers: `Conv1`, `MaxPool`, `Flatten`, and `Dense2`.

   (d) (1 point) Your dataset has 100,000 images with approximately 20 images per person. Describe briefly how you would partition this dataset into train and validation sets (a separate test set exists).

   (e) (2 points) After your very first training attempt, you find out that both train and validation accuracies are low, but close to each other. What can you do to improve your model?

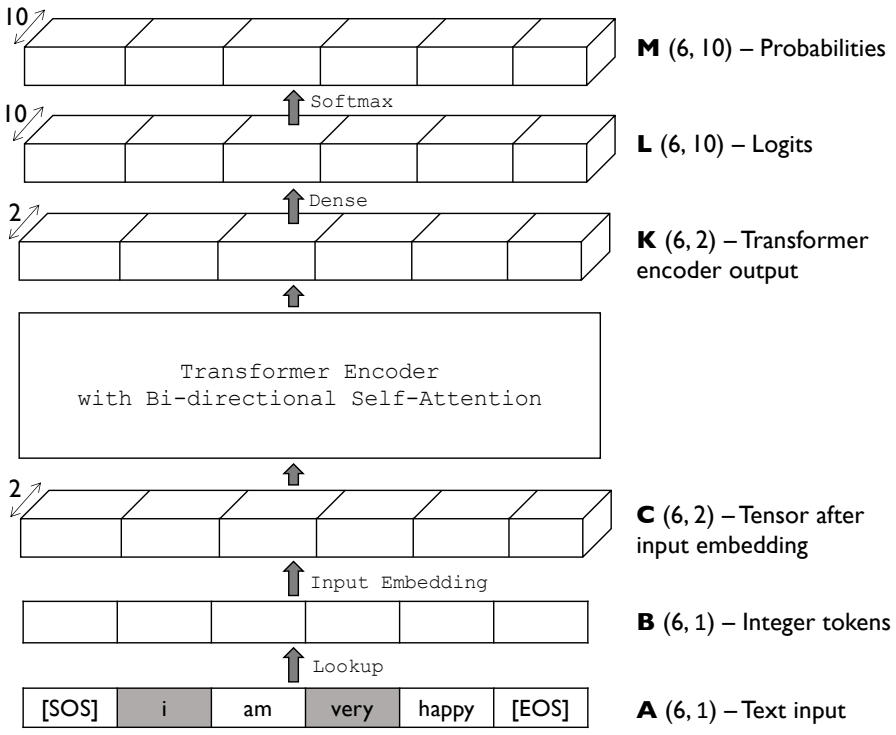
   (f) (3 points) After a few development rounds, you have a trained model that works well on both train and validation datasets. Your client tests it on their test set and says it works well on test images taken under good lighting conditions, but fails miserably on test images taken under mildly poor lighting conditions. What could be a reason? How can you make your model work on poorly-lit images as well?

(g) (4 points) Your client now asks you to recognize faces of 10,000 new people. You have 5 labelled images for each person in this new dataset. Briefly describe how you can re-purpose the same network you trained before (without re-training its weights) to solve this new task.

3. (20 points) You are building a toy neural network to better understand transformer-based models. To keep things simple, you convert all text inputs to lowercase and tokenize them using white spaces. Your vocabulary has a few English words, and three special tokens: [Mask], [SOS], and [EOS]. They denote **Mask**ed-out positions, **S**tart **O**f **S**entence positions, and **E**nd **O**f **S**entence positions, respectively. Your entire vocabulary is shown below with the input embedding vector for each token at the start of training.

Word (Token)	Token ID	Initial Input Embedding
[Mask]	0	[0.2, 0.5]
[SOS]	1	[0.8, 1.3]
[EOS]	2	[0.1, 0.6]
am	3	[0.7, 0.3]
angry	4	[2.0, 1.3]
happy	5	[0.7, 0.3]
i	6	[0.9, 0.6]
not	7	[0.9, 0.4]
sad	8	[0.6, 1.9]
very	9	[0.1, 0.3]

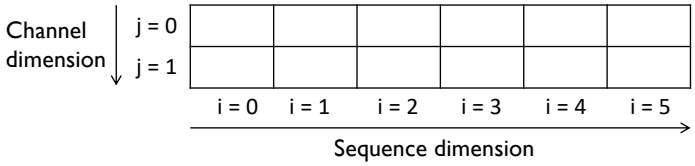
The architecture of your network is sketched below with an example input. Tensor names (**A**, **B**, *etc.*) are on the right. The shape of each tensor is shown next to its name using the convention (`sequence_length`, `channels`) (the batch dimension has been ignored). The network is initially set up for masked language modeling (MLM) based pre-training. A fixed sequence length 6 is used (token sequences are either cropped or padded to keep the sequence length at 6).



(a) You are going to first pre-train the network shown above with MLM. The very first sentence input is: "i am very happy", with the tokens "i" and "very" randomly chosen to be masked out. We now analyze the forward pass for this input.

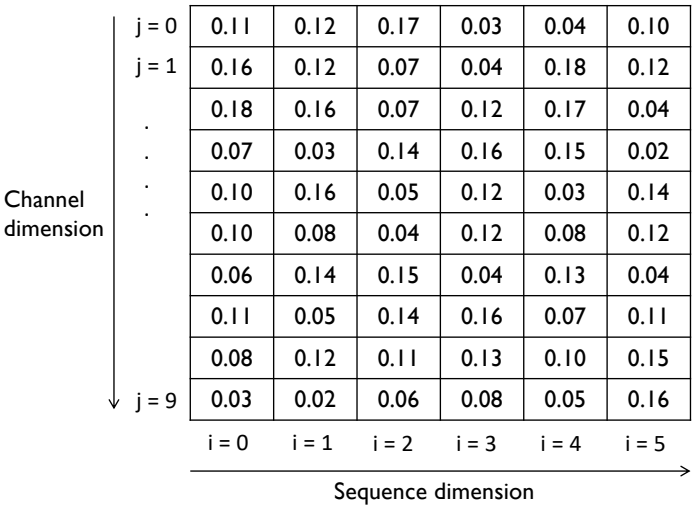
i. (2 points) Write down the contents of the integer tensor **B**.

ii. (4 points) Copy the following tensor **C** to your answer sheet and complete its values for the above input. The input embeddings are shown in the vocabulary table above.



iii. (2 points) The `Dense` layer between the tensors **K** and **L** changes the channel dimension length from 2 to 10 while applying the same fully-connected operation at each location of the sequence. How many learnable parameters are there in this dense layer?

iv. (5 points) For the above input, the probabilities at the end of the network (contents of tensor **M**) are shown below. Write down the value of the total loss for this datum. (**Hint**: The total loss is the average of the cross-entropy losses at the masked locations.)



(b) (4 points) After MLM pre-training, you decide to fine-tune the network for sentence sentiment classification. To obtain an embedding for the whole sentence, you plan to remove everything after tensor **K** and pool **K** across the sequence dimension. State two sufficiently different pooling strategies that you can use.

(c) (3 points) You decide to also try next-token prediction pre-training (aka language model pre-training as done in GPT-style networks). Briefly mention one key change you need to make to the transformer encoder block in the diagram when you pre-train with this objective.

Definition 1. *(Expected value): Let* $\mathrm{X} \in \{x_1, x_2, \ldots\}$ *be a discrete random variable with the probability mass function* $p(x)$*. Then the expected value of a function* $g(\mathrm{X})$ *of* X *is,*

$$\mathrm{E}[g(\mathrm{X})] = \sum_{i=1}^{\infty} g(x_i)\, p(x_i).$$

Definition 2. *(Cross-entropy loss): Let* $\mathbf{p} = (p_1, p_2, \ldots, p_{\mathrm{L}})$ *be the output probabilities produced by a model for a given datum and let* $y \in \{1, 2, \ldots, \mathrm{L}\}$ *be the groundtruth label for that datum. The cross-entropy loss is then given by*

$$l(y, \mathbf{p}) = -\log(p_y).$$

The following results hold when X, Y are random variables, a, b are constants, and g, h are functions.

1. $V[X] = E[X^2] - E[X]^2$
2. $E[aX + b] = aE[X] + b$
3. $V[aX + b] = a^2 V[X]$
4. $E[X + Y] = E[X] + E[Y]$
5. $V[X + Y] = V[X] + V[Y] + 2\,\mathrm{cov}(X, Y)$
6. $E[g(X)\,h(Y)] = E[g(X)]\,E[h(Y)]$, when X and Y are independent
7. $\mathrm{cov}(X, Y) = E[XY] - E[X]E[Y]$

① (a) Because $2Y$ & $Y_1 + Y_2$ have the same distribution,

$$V[2Y] = V[Y_1 + Y_2]$$

(independent) $\rightarrow 0$

$$4V[Y] = V[Y_1] + V[Y_2] + 2\,\mathrm{Cov}(Y_1, Y_2)$$

$$4V[Y] = 2V[Y]$$

But according to the problem statement, $V[Y]$ is not zero.

$\therefore V[Y]$ must be infinite.

(b) (i) $$L(\theta) = \prod_{i=1}^{n} \theta(1-\theta)^{k_i - 1}$$

(ii) $$\log L(\theta) = \sum_{i=1}^{n} \{\log\theta - (k_i - 1)\log(1-\theta)\}$$

Setting the derivative to zero gives,

$$n\frac{1}{\theta} = \sum_{i=1}^{n} \frac{k_i}{(1-\theta)} - n\frac{1}{(1-\theta)}$$

$$n - n\theta = \theta\sum k_i - n\theta$$

$$\theta = \frac{n}{\sum k_i}$$

$$\therefore \hat{\theta}_n = \frac{n}{\sum_{i=1}^{n} k_i}$$

(iii)

$$E[\hat{\theta}_1] = E[1/k_1] \quad \text{with} \quad k_1 \text{ distributed as } X.$$

$$\therefore E[1/k_1] = E[1/X] = \sum_{k=1}^{\infty} \frac{1}{k} \Pr(X=k) = \sum_{k=1}^{\infty} \frac{1}{k} \theta(1-\theta)^{k-1}$$

$$= \theta + \frac{1}{2}\theta(1-\theta) + \frac{1}{3}\theta(1-\theta)^2 + \cdots$$

(iv) The estimator is biase because:

$$E[\hat{\theta}] \neq \theta.$$

(2) (a) $L = 5000$ (the number of classes).

(b) B : (128, 64, 32)

C : (32, 16, 32)

(c) Conv1 : $3 \times 3 \times 3 \times 32 + 32$

MaxPool : 0

Flatten : 0

Dense2 : $128 \times L + L$

(d) 20% of the images of each person in the val set. Or any other sensible answer.

(e) Use a bigger model, train for longer, use a better optimizer, any method that can reduce bias.

Increasing the train set size won't work so that answer is not accepted.

(f) Possible reason: Train (& validation) sets do not have images taken under poor lighting conditions.

Fixes:

1. Use data augmentation during training to make the model robust to brightness / contrast / lighting changes. -OR-
2. Normalize inputs. This will reduce the sensitivity to illumination changes.

(g) Remove everything after tensor F & use the network as a feature extractor. Then do a nearest neighbor search.

(3) (a)

(i)

1	0	3	0	5	2

(ii)

0.8	0.2	0.1	0.2	0.7	0.1
1.3	0.5	0.6	0.5	0.3	0.6

(iii) 2×10 or $2 \times 10 + 10$

(iv) $\frac{1}{2}\left(-\log \text{Prob}(i=1, j=6) - \log \text{Prob}(i=3, j=9)\right)$

$= \frac{1}{2}(-\log 0.14 - \log 0.08)$

Masked locations

Groundtruth IDs at the masked locations.

(b) One from each category below:

1. CLS/SOS/EOS pooling. (taking the first/last embedding.)
2. Average/min/max pooling.

(c) Remove forward looking connections from the bi-directional self-attention blocks.