CS-461

Foundation Models and Generative AI

Exam Recap Session

Charlotte Bunne, Fall Semester 2025/26

Exam Logistics

Friday 16.01.2026 from 15:15 to 18:15 in CE 1 515 and PO 01.

Closed book exam. One A4 crib sheet (both sides).

Questions and Content

The exam will feature content from **all classes, guest lectures and tutorials.**

The exam is a **multiple choice exam**.

There are three types of questions in the exam.

1 - **Factual knowledge questions** about topics covered in classes, guest lectures and tutorials.

2 - **Math questions** that either ask you to **a)** do basic computations without guidance, or **b)** read through more advanced proofs and spot eventual errors or missing arguments.

3 - **Code questions** that ask you to proof-read short code blocks.

Structure

The exam has five parts.

Part I consists of a set of **20 factual knowledge** questions that require no computations. All questions are independent.

Parts II-V focus on 3 different topics related to subjects covered in classes, guest lectures and tutorials. Questions in these parts may not be independent.

Required Code and Math Knowledge

Code

The exam requires basic knowledge of **Python** and **PyTorch**. The blocs of code that you will have to examin are short (less than 20 lines) and self-contained.

Math

The exam requires **undegraduate knowledge of probability, analysis and statistics**. No excessively long, creative or uncommon computations are required.

Graduate level topics and objects, when needed, are fully detailed and introduced at the beginning of each part.

Example - Factual Knowledge Questions

In self-supervised learning, what is the main difference between contrastive learning and masked learning?

See Lecture 2 - Learning at Scale: Supervised, Self-Supervised and Beyond

Example - Factual Knowledge Questions

Masked vs. Contrastive Learning. In self-supervised learning, what is the main difference between contrastive learning and masked learning?

Answer A. Contrastive learning requires labeled negative examples, while masked learning is fully unsupervised.

Answer B. Contrastive learning learns representations by comparing pairs of inputs and optimizing relative similarity between positive and negative examples, whereas masked learning predicts missing parts of an input from its observed context.

Answer C. Masked learning optimizes mutual information directly, whereas contrastive learning optimizes likelihood.

Answer D. Contrastive learning can only be applied to images, while masked learning is specific to language and sequential data.

Example - Factual Knowledge Questions

Masked vs. Contrastive Learning. In self-supervised learning, what is the main difference between contrastive learning and masked learning?

Answer A. Contrastive learning requires labeled negative examples, while masked learning is fully unsupervised.

Answer B. Contrastive learning learns representations by comparing pairs of inputs and optimizing relative similarity between positive and negative examples, whereas masked learning predicts missing parts of an input from its observed context.

Answer C. Masked learning optimizes mutual information directly, whereas contrastive learning optimizes likelihood.

Answer D. Contrastive learning can only be applied to images, while masked learning is specific to language and sequential data.

Example - Math Questions

Properties of Softmax Functions. Recall that the softmax function is defined as

$$\phi_\tau : \mathbf{x} \in \mathbb{R}^d \mapsto \left[\frac{e^{\mathbf{x}_j \tau^{-1}}}{\sum_{i=1}^{d} e^{\mathbf{x}_i \tau^{-1}}} \right], \quad j = 1, \ldots, d\,.$$

Basic Properties. Which of the following statements is true?

Answer A. The SoftMax function is invariant by translation i.e. $\phi_\tau(\mathbf{x} + \lambda) = \phi_\tau(\mathbf{x})$.

Answer B. The SoftMax function is differentiable only if at least one coordinate of $\mathbf{x}$ is non-zero.

Answer C. The SoftMax function is invariant to multiplication of $\mathbf{x}$ by a scalar i.e. i.e. $\phi_\tau(\mathbf{x} \cdot \lambda) = \phi_\tau(\mathbf{x})$.

Answer D. The SoftMax function approaches the one-hot argmax function as the temperature goes to $+\infty$.

Example - Math Questions

Properties of Softmax Functions. Recall that the softmax function is defined as

$$\phi_\tau : \mathbf{x} \in \mathbb{R}^d \mapsto \left[\frac{e^{\mathbf{x}_j \tau^{-1}}}{\sum_{i=1}^{d} e^{\mathbf{x}_i \tau^{-1}}} \right], \quad j = 1, \ldots, d\,.$$

Basic Properties. Which of the following statements is true?

Answer A. The SoftMax function is invariant by translation i.e. $\phi_\tau(\mathbf{x} + \lambda) = \phi_\tau(\mathbf{x})$.

Answer B. The SoftMax function is differentiable only if at least one coordinate of $\mathbf{x}$ is non-zero.

Answer C. The SoftMax function is invariant to multiplication of $\mathbf{x}$ by a scalar i.e. $\phi_\tau(\mathbf{x} \cdot \lambda) = \phi_\tau(\mathbf{x})$.

Answer D. The SoftMax function approaches the one-hot argmax function as the temperature goes to $+\infty$.

Example - Math Questions

Properties of Softmax Functions. Recall that the softmax function is defined as

$$\phi_\tau : \mathbf{x} \in \mathbb{R}^d \mapsto \left[\frac{e^{\mathbf{x}_j \tau^{-1}}}{\sum_{i=1}^d e^{\mathbf{x}_i \tau^{-1}}} \right], \quad j = 1, \ldots, d\,.$$

Basic Properties. Which of the following statements is true?

Answer A. We have $\left\| \phi_\tau(\mathbf{x}) \right\|_1 = \tau$ for all $\mathbf{x} \in \mathbb{R}^d$.

Answer B. We have $\left\| \phi_\tau(\mathbf{x}) \right\|_1 > 1$ for all $\mathbf{x} \in \mathbb{R}^d$ and $\tau > 0$.

Answer C. We have $\left\| \phi_\tau(\mathbf{x}) \right\|_1 = 1$ for all $\mathbf{x} \in \mathbb{R}^d$ and $\tau > 0$.

Answer D. We have $\arg \max_{i=1,\ldots,d} \left[\phi_\tau(\mathbf{x}) \right]_i = \arg \min_{i=1,\ldots,d} \mathbf{x}_i^2$ for all $\mathbf{x} \in \mathbb{R}^d$.

Example - Math Questions

Properties of Softmax Functions. Recall that the softmax function is defined as

$$\phi_\tau : \mathbf{x} \in \mathbb{R}^d \mapsto \left[\frac{e^{\mathbf{x}_j \tau^{-1}}}{\sum_{i=1}^d e^{\mathbf{x}_i \tau^{-1}}} \right], \quad j = 1, \ldots, d\,.$$

Basic Properties. Which of the following statements is true?

Answer A. We have $\left\| \phi_\tau(\mathbf{x}) \right\|_1 = \tau$ for all $\mathbf{x} \in \mathbb{R}^d$.

Answer B. We have $\left\| \phi_\tau(\mathbf{x}) \right\|_1 > 1$ for all $\mathbf{x} \in \mathbb{R}^d$ and $\tau > 0$.

Answer C. We have $\left\| \phi_\tau(\mathbf{x}) \right\|_1 = 1$ for all $\mathbf{x} \in \mathbb{R}^d$ and $\tau > 0$.

Answer D. We have $\arg \max_{i=1,\ldots,d} \left[\phi_\tau(\mathbf{x}) \right]_i = \arg \min_{i=1,\ldots,d} \mathbf{x}_i^2$ for all $\mathbf{x} \in \mathbb{R}^d$.

Example - Code Question

We have trained a variational auto-encoder. The training procedure yields us an `encoder` and a `decoder` network.

Example - Code Question

We have trained a variational auto-encoder. The training procedure yields us an `encoder` and a `decoder` network.

Which of the following PyTorch codes correctly implements **sampling** from this model?

```
class VAE(nn.Module):
    def __init__(self, encoder, decoder):
        self.denoiser = denoiser
        self.decoder = decoder

    def sample(self, T, shape):
        x = randn(shape)
        for t in reversed(range(1, T + 1)):
            mu, log_var = self.denoiser(x, t)
            x = mu + exp(0.5 * log_var) * randn_like(x)
        return x
```

```
class VAE(nn.Module):
    def __init__(self, encoder, decoder):
        self.encoder = encoder
        self.decoder = decoder

    def sample(self, n, device):
        z = randn(n, self.z_dim, device=device)
        x_hat = self.decoder(z)
        return x_hat
```

```
class VAE(nn.Module):
    def __init__(self, encoder, decoder):
        self.encoder = encoder
        self.decoder = decoder

    def sample(self, x):
        mu, logvar = self.encoder(x)
        z = mu + exp(0.5 * logvar) * randn_like(mu)
        return self.decoder(z)
```

```
class VAE(nn.Module):
    def __init__(self, model, vocab_size):
        self.encoder = encoder
        self.vocab_size = vocab_size

    def sample(self, max_len, bos_id):
        tokens = [bos_id]
        for t in range(max_len - 1):
            logits = self.model(tokens)
            next_id = sample_categorical(softmax(logits[-1]))
            tokens.append(next_id)
        return tokens
```

Example - Code Question

We have trained a variational auto-encoder. The training procedure yields us an `encoder` and a `decoder` network.

Which of the following PyTorch codes correctly implements inference using this model?

```
class VAE(nn.Module):
    def __init__(self, encoder, decoder):
        self.denoiser = denoiser
        self.decoder = decoder

    def sample(self, T, shape):
        x = randn(shape)
        for t in reversed(range(1, T + 1)):
            mu, log_var = self.denoiser(x, t)
            x = mu + exp(0.5 * log_var) * randn_like(x)
        return x
```

```
class VAE(nn.Module):
    def __init__(self, encoder, decoder):
        self.encoder = encoder
        self.decoder = decoder

    def sample(self, x):
        mu, logvar = self.encoder(x)
        z = mu + exp(0.5 * logvar) * randn_like(mu)
        return self.decoder(z)
```

```
class VAE(nn.Module):
    def __init__(self, encoder, decoder):
        self.encoder = encoder
        self.decoder = decoder

    def sample(self, n, device):
        z = randn(n, self.z_dim, device=device)
        x_hat = self.decoder(z)
        return x_hat
```

```
class VAE(nn.Module):
    def __init__(self, model, vocab_size):
        self.encoder = encoder
        self.vocab_size = vocab_size

    def sample(self, max_len, bos_id):
        tokens = [bos_id]
        for t in range(max_len - 1):
            logits = self.model(tokens)
            next_id = sample_categorical(softmax(logits[-1]))
            tokens.append(next_id)
        return tokens
```

CS-461

Foundation Models and Generative AI

Good luck for the exam!

Questions?