

Quiz#3
B6639334
นางสาวพิมพ์นารา อุดุลจันทรศร

1.)

1.2 Pre-trained model หมายถึงโมเดลที่มีคุณลักษณะอย่างไร

ตอบ คุณลักษณะเด่น 1. ผ่านการฝึกหัดด้วยข้อมูลขนาดใหญ่ (Massive Dataset Training) 2. ใช้เทคนิค Transfer Learning (การนำร่องการเรียนรู้) 3. มีความสามารถในการทบทวน (Generalization)

1.4 การ fine-tune โมเดลทาง Deep Learning คืออะไร

ตอบ คือ กระบวนการที่นำโมเดลที่ผ่านการฝึกหัดมาฝึกอีกครั้ง (Retrain) ด้วยข้อมูลเฉพาะทาง เพื่อให้โมเดลนั้นมีคุณสมบัติใหม่ๆ ในงานที่เราต้องการโดยเฉพาะ

1.5 Learning rate คืออะไร หากกำหนดค่า learning rate สูงหรือต่ำเกินไปจะส่งผลอย่างไรต่อการเรียนรู้ของโมเดล

ตอบ Learning Rate คือ ค่าที่กำหนดว่า ที่ใดในการคำนวณหนึ่งรอบ (Step size) ในการปรับปูโมเดลแต่ละคันเพื่อให้โมเดลวิเคราะห์ความแย่ร้ายและทำให้ค่าความผิดพลาด (loss) น้อยลง ฟรีชิบเพื่อเข้าใจการเดินทางจากภูมิภาคไปยังภูมิภาคอื่นๆ ดังนี้ 1. ภูมิภาค คือ ค่าความผิดพลาด (Loss function)

2. จุดต่ำสุดของภูมิภาค คือ จุดที่ไม่มีผลลัพธ์ต่อไป (Global Minimum)

3. Learning Rate คือ ความถูกของการเดินทางในภูมิภาค

หากกำหนด Learning Rate สูงเกินไป ส่งผลให้ Model จะกระโดดข้ามจุดต่ำสุดไปทางทิศทาง Loss จะแทรกซึ้งลง จนกว่า error ต่ำสุดขึ้น

หากกำหนด Learning Rate ต่ำเกินไป ส่งผลให้ Model เสียเวลา มาก Loss จะลดลงแบบช้าๆ

1.6 Vanishing gradient คืออะไร ทำไมถึงมีเป็นปัญหา กับโมเดล deep learning

ตอบ Vanishing gradient คือ ปัญหาที่เกิดขึ้นเมื่อ Model วิ่งผ่านชั้นของ Neural Network มาแล้วสักรอบค่า weight ของชั้นก่อนจะหายไปซึ่งเราไม่สามารถคำนวณค่าใน

การปรับค่า weight ของชั้นก้าวไปซึ่งหมายความว่าการปรับค่า weight ของชั้นก้าวไม่ได้ปรับ

Vanishing gradient พบว่ามากับ Deep learning เพราะ Early layers Freeze

1.7 Word Embedded คืออะไร

ตอบ คือ เทคนิคใน NLP (Natural Language Processing) ที่แปลงคำศัพท์ให้เป็นตัวเลข (Vector) เพื่อให้คอมพิวเตอร์นำไปประมวลผลและใช้ความหมายได้

1.8 เทคโนโลยี GAN และ LLM จึงจัดอยู่ในกลุ่มของโมเดลแบบ Generative AI

ตอบ เหตุผลสำคัญของ GAN และ LLM คือ การสร้างข้อมูลใหม่ ไม่ใช่แค่การรีเคป หรือจัดหมวดหมู่ข้อมูลเดิม

2.)

2.1 Backpropagation คืออะไร และมีบทบาทอย่างไรในกระบวนการเรียนรู้ของ neural network

ตอบ คือ กระบวนการปรับค่า weight ของชั้นก่อนหน้าไปต่อไป 1. ระบุถูกต้องไปต่อไปเรื่อยๆ : Neural network วิ่งรอบไปเรื่อยๆ ที่เราไม่รู้จะไปต่อไปอย่างไร 2. Backprop ฟันว่าที่เราต้องการปรับค่า Gradient 3. Backprop เชื่อมโยงผลลัพธ์สุดท้ายเหตุ : ผ่านส่วนต่อประสานของหัวใจต่อสู่หัวใจต่อไป ระหว่างผลลัพธ์ที่ได้มา

กับข้อมูลที่ต้องการ ทำให้ Model ค่อนข้างดูน่าเชื่อถือในทุกรอบการทำงาน (Epoch)

2.2 โมเดล GAN (Generative Adversarial Network) จัดว่าเป็นโมเดลแบบ Generative AI เพราะอะไร จึงอธิบายแนวคิดของโมเดล GAN และความสัมพันธ์ระหว่างแนวคิดของโมเดล GAN กับแนวคิดของ Generative AI ให้เข้าใจให้ได้มากที่สุด

ตอบ GAN คือ Genetive AI เพราะ ทำนายอย่างสุ่ม คือ การสร้างข้อมูลใหม่ไม่ใช่การวิเคราะห์ข้อมูลที่มีอยู่แล้ว

แนวคิดของ GAN คือการนำ Neural Network 2 ตัวๆ แข่งกันเอง โดยฝ่ายละฝ่ายแพ้ 2 ตัว ฝ่ายที่

1. The Generator (G) - ผู้ผลิตภาพ (ผู้สร้าง)

- ต้องเรียนรู้เพื่อให้มีภาพตามคนที่ต้องการโดยไม่รู้ว่าในภาพนั้นเด็กนั้นตาดูตรงไปทางซ้าย - เป้าหมาย : พยายามสร้างภาพที่ใกล้

2. The Discriminator (D) - ຕ່າງໆທີ່ສູນ໌ນລັກງານ (ຜູ້ແຍກເບີຍ)

- ข้อมูลที่มี : ตำแหน่งนั่งแบบต่อต้องอยู่ในฟอร์ม (ข้อมูลสำหรับ train)

- น้ำที่ : อยู่รับแบ่งกันเพื่อท้าท่วมแบ่งต่อไป และแบ่งต่อไปอีก (ที่ Generator ส่องว่า) แล้วต่อสิ่งที่แบ่งต่อให้จริง

- เป้าหมาย: จับผิดแบบค์ป้องวิให้ได้มากที่สุด

2.4 Reinforcement Learning แตกต่างจาก Supervised Learning อย่างไร จงอธิบายองค์ประกอบหลักของ Reinforcement Learning และยกตัวอย่างการประยุกต์ใช้งานจริง

ตัวอย่าง Reinforcement คือ การเรียนรู้แบบ込むฝึกด้วยผู้สอน (Supervised) คือ การเรียนรู้แบบมีผู้สอน (มีเคล็ดลับชี้แจงอยู่)

องค์ประกอบหลักของ Reinforcement Learning ประกอบด้วย 5 ส่วน

1. Agent (អង្គរោះ) : កើត ពាក្យ AI នូវ Model មួយនេះ ដែលត្រួតពិនិត្យការងារ

2. Environment (สิ่งแวดล้อม) : ตัว โลโก้ Agent อาศัยอยู่ เช่น กรอบหน้าจอ

3. State (สถานะ) : คือ สถานะ-การณ์ปัจจุบันที่ Agent ของเรานั้น เช่น ตอนนี้ศัตรูอยู่

4. Action (การกระทำ): ต้องสื่อว่า Agent เลือกทำได้ในสถานะปัจจุบัน เช่น การโกรล์

5. Reward (報酬) : ถ้า ผลลัพธ์ของ Environment ในทักษะทางเพื่อทักษะงานน้ำ ทำได้ดี ก็จะได้

ຕົວອັນການປະຍາດໄສງານຈົນ

1. การเล่นเกม 2. หน่วยเรียนรู้ 3. ภาคผนวกที่ 4. ระบบแผนผาด้วยภาษาไทย

2.5 Agentic AI คืออะไร จงอธิบายความแตกต่างระหว่าง Agentic AI และ LLM แบบดังเดิมในเงื่อนไข

ຄອງ Agentic AI ດີ່ນໄດ້ພາຍໃນເຫັນວ່າ AI ຊື່ສື່ ຂະແໜນ ແລະ ລວມມື້ນຳ ເທົ່ານັ້ນມາຫຍຸ້ງຕ່າງໆໂລຢີໄສ້ຮັບຮູ້ອຳນັດສິ່ງວ່ານີ້

ຄ່າມຕິດວັນ Agentic AI ມີລັບ LLM 1. Agentic AI

↳ វីរាមាននៃទំនាក់ទំនងគ្រប់គ្រង់ទៅលក្ខណៈ (Sust - task)

↳ Reflective : คุณครูต้องหันกลับมารักษาตัวเอง (Self-reflection)

2 11 M

↳ ตัวอย่างเช่น: ทั่วไปแบบ input → ဝรรษ์อ่า จนินทร์กวนเด็ก

↳ ໃຊ້ ສິ່ງທີ່ມີມາແລ້ວ : ເພີ້ມໄກຕະຫຼອງດຳລົງທີ່ມີມາຈຸດກັບລູກຂຶ້ນ (Pre-trained data)

↳ Static : ជាការណែនាំត្រូវបានរក្សាទេ ហើយត្រូវបានរក្សាទេដោយការងារ (program) ។

2.6 เหตุใด Transformer จึงมีความสามารถในการจัดการ long-range dependency ได้ดีกว่า RNN แบบดั้งเดิม ลองอธิบาย

ต่อ ห้องเรียนของ RNN คือ รีบ input ของ Model จะรีบ input แรกๆ กล่าวคือ RNN: ห้องเรียนจะก้าวเดินต่อไปจนถึงเวลาที่เรียกว่า Hidden State ในยังต่อที่ 2, 3, ..., ไม่ใช่อย่างที่ Transformer ไม่ได้อ่าน input ที่: แต่ผู้อ่านต้องมีพื้นฐานของ และที่สำคัญ Self-Attention

ใน Self-Attention ผู้รับสารจะหาตัวที่เกี่ยวข้องกัน แล้วคำนวณค่าของตัวที่เกี่ยวข้อง (Attention Score) ให้กับตัวอื่นๆ ได้โดยตรง อีสาน: ต่อตัวกัน

ข้อดี: ไม่ต้องแยกกับตัวสืบต่อสาย: นำเข้าแต่ไฟในท่อเดียวของ transformer 5:50: ใช้งานง่ายขึ้นและต่อกัน 1 เมตร

2.9 เหตุใด Convolutional Neural Network (CNN) จึงเหมาะสมกับงานด้านการประมวลผลภาพมากกว่า Neural Network แบบ Fully Connected (นั่นคือ MLP) จังอธิบาย

ຕອນ ដែលមាន MLP គោរពការ MLP ការងារទូទៅនៃការងារទាំងអស់ វាត្រូវបានចូលទៅ MLP ដើម្បី ហើយ MLP រួម input ដែល ជាទិន្នន័យ (Vector) ដែល

CNN รับ input ที่มีก้อน 2 มิติ (Matrix) โดยตรงแล้วให้ Filter (ผ่านกรอง) แล้วแปลงเป็นภาพ ผลลัพธ์: ลักษณะเดียวกับ pixel ที่อยู่ติดกันด้วย pixel ใน input ทำให้เก็บ feature เช่น เส้นขอบ บุคคล หรือตัวอักษรที่มีความซับซ้อน

3.)

- (b). Learning Algorithms หรือ Optimization Algorithms คืออะไร ยกตัวอย่าง Learning Algorithm ที่นิยมใช้ในการเทรนโมเดล Deep Learning มา 2 Algorithms พร้อมอธิบายในรายละเอียดของแต่ละ Algorithm คร่าวๆ

ตอบ Learning Algorithms หรือ Optimization Algorithm คือ กลไกทางคณิตศาสตร์ ที่ทำหน้าที่ปรับเปลี่ยนค่าทางวัสดุ (Weight & Bias) ของ Model ในขั้นตอน train ให้ลดลง loss มากที่สุด

1. Stochastic Gradient Decent (SGD)

↳ หากพิจารณาข้อมูลทั้งหมด (Full Batch) SGD จะใช้วิธี "สุ่มข้อมูลมาแต่ละรอบ" (Mini-batch) เพื่อตัดตอน แล้วกาก่อน

1. สุ่มนิยมข้อมูลมาลุ่มเล็กๆ (เช่น 32 หรือ 64 ภาค)

2. คำนวน Gradient (ความชัน) ตามข้อมูลลุ่มนี้

3. อัปเดต Weight ใหม่ ดังนี้ $W_{\text{new}} = W_{\text{old}} - \text{Learning Rate} \times \text{Gradient}$

4. วนการซ้ำไปเรื่อยๆ

2. Adam (Adaptive Moment Estimation)

↳ ปรับปรุง SGD ให้ดีขึ้น

หลักการทำงาน : Adam ไม่ได้อ่านทั้ง ตามชั้นปัจจุบัน แล้วใช้การนับ สถิติ 2 อย่างไว้ในท้องตามา

1. Momentum : จำลองการทำงานเดิมๆ ผ่านชั้นปัจจุบัน แล้วนำมารักษาไว้ไปทางเดิมเรื่อยๆ

2. Adaptive Learning Rate : ถูกออกแบบให้ในชั้นปัจจุบัน รักษาขนาดตัวค่า ส่วนต่อไปที่จะเปลี่ยนแปลงตัวค่าให้มีน้อยลง

- (d). Confusion Matrix นิยมนำมาใช้แสดงผลลัพธ์ของอะไร แล้วค่า Accuracy, Precision, และ Recall สามารถมาจากการ Confusion Matrix ได้อย่างไร แต่ละค่าดังกล่าวสื่อความหมายอย่างไร จงอธิบาย หรือยกตัวอย่างประกอบ

ตอบ Confusion Matrix นิยมนำมาแสดงผลลัพธ์ของ Classification Model ในการเรียนรู้แบบฟังผู้สอน

ตัวอย่าง สร้าง矩阵สำหรับการเรียนรู้ปัญญาประดิษฐ์ - 1

• Positive : ผู้ป่วย

• Negative : ไม่เป็นโรค

1. True Positive (TP) : กรณี "ป่วย" ผลจริงๆ ก็ "ป่วย" (กรณี - ผู้ป่วยที่จริง)

2. True Negative (TN) : กรณี "ไม่ป่วย" ผลจริงๆ ก็ "ไม่ป่วย" (กรณี - ไม่เป็นโรคทั่วไป)

3. False Positive (FP) : กรณี "ป่วย" ผลจริงๆ ไม่ป่วย (กรณี +)

4. False Negative (FN) : กรณี "ไม่ป่วย" ผลจริงๆ ตันป่วย (กรณี -)

การคำนวณ-ตัวต่อตามนี้ (Accuracy, Precision, Recall)

$$1. \text{Accuracy} (\text{ความถูกต้อง}) = \frac{TP+TN}{\text{Total}}$$

$$2. \text{Precision} (\text{ตัวความแม่นยำของผู้ตรวจ}) = \frac{TP}{TP+FP}$$

$$3. \text{Recall} = \frac{TP}{TP+FN}$$

- (f). อธิบายความแตกต่างระหว่าง Batch Gradient Descent, Stochastic Gradient Descent (SGD) และ Mini-batch Gradient Descent พร้อมอธิบายข้อดี-ข้อจำกัดของแต่ละวิธี

ตอบ ความต่างหลักทั้ง 3 วิธีนี้ อยู่ที่ ปริมาณข้อมูล ที่นำมาใช้ในการคำนวณเพื่อปรับแต่ง Weight

1. Batch Gradient Decent

↳ นำเอาข้อมูลทั้งหมดในสูตรเข้ามาร่วมคำนวณให้แล้ว คำนวณฝั่งผลลัพธ์ (Error) ก่อนแล้วถึงขั้นตอน 1 ขั้น

ข้อดี : 1. ทราบได้เรียบเนียน : เน้นการสู่จุดที่สุดคุณภาพและเสถียรมากกว่า Loss ด้วยผลลัพธ์ที่แม่นยำ

2. เมื่อนำร่อง : หา Gradient สำหรับความซับซ้อนของข้อมูลทั้งหมด

ข้อจำกัด : 1. รักษา : รักษาข้อมูลมาได้ชั่วโมงคุ้มค่า

2. ทรัพยากรห้องแม่ข่าย : ต้องโอนลากข้อมูลทั้งหมดลงมาในหน่วยความจำที่จำกัด

2. Stochastic Gradient Descent - SGD

↳ สุ่มขึ้นข้อมูล 1 ตัวอย่าง แล้วคำนวณ Error

ข้อดี : 1. รับโอกาสเร็ว : ไม่ต้องรอข้อมูลทั้งหมด

2. ไม่มี memory อยู่ : ไม่ต้องจดจำที่ลืมตัว

ข้อเสีย : 1. เน้นการรุ่งเรือง : การคำนวณ Loss จะเป็นชิ้นๆ รวมถึงทำให้ขยำที่สุด ข้อดีที่สุด (มีงานนี้มากอยู่ในบัญชีต่อต่อต่อ)

2. ใช้ Hardware ไม่ดี : ใช้ GPU ไม่เต็มที่

3. Mini batch Gradient Descent

↳ ใช้มีพื้นที่จดจำข้อมูลให้คุ้มค่า โดยนำเข้ามาใน 1 Batch ขนาดต่ำกว่า แล้วคำนวณ Weight 1 ครั้ง

ข้อดี : 1. สมดุลตัวต่อต่อ : เลือกใช้ร่วมกับ SGD และเลือกใช้ Batch Gradient Descent

2. ปรับเปลี่ยน Hardware ง่ายๆ : หาก Batch size มีขนาดเหมาะสมกับ GPU

ข้อจำกัด : ต้องดู Hyperparameter : ต้องหากำหนดค่า Batch size ใจ

- (g). อธิบายแนวคิดของ Attention Mechanism และเหตุใดกลไกนี้จึงช่วยให้โมเดลอย่าง Transformer สามารถจัดการข้อมูลลำดับ (sequence) ได้ดีกว่า RNN แบบเดิมๆ

ตอบ แนวคิดของ Attention "เลือกคำเฉพาะ: ถูกหันกลับ" โดยของ Model ไว้ไม่ต้องพยายามจำทุกอย่างข้อมูลในที่เดียวแล้วให้กลับไปเก็บคุ้มข้อมูลอันฉบับไปแล้วตามเดิม

เหตุผลที่ Transformer ใช้ Attention ต้องมากกว่า RNN 1. ไม่เน้น Long-Range Dependency

↳ ลักษณะ Self-Attention ทุกคำให้ไปรู้เรื่องตัวเองว่ามันจะมี impact บนคำไหน

- ยกตัวอย่าง "ค่าน้ำ" กับ "คำที่ 10,000" คำที่หันกลับ 1 (พื้นที่หันกลับ)

- กรณี Model พิจารณาที่หันกลับแล้วก็มองไปที่หันกลับ

2. ภาษา: ภาษาและแบบหน้า

↳ รับข้อมูลเข้ามาทั้งประโยค ทั้งที่หันกลับแล้ว ผ่านไป Attention คำไหนควรส่วนหนึ่งที่ถูกหันกลับ

- ผลลัพธ์ Transformer มากกว่า RNN หลายเท่า

3. ความสำคัญในภาษาต่อๆ กัน

↳ สามารถหันกลับ Attention Weights บน heatmap ให้ดูว่า "ตอนที่ AI บุคลากรหันกลับจะไปที่ตัวไหน" ทำให้เราสามารถหาตัวของ AI ได้ดีขึ้น

สรุป Attention ไม่ใช่แค่ตัว "ทำจากอย่างเดียว" (RNN) anymore "ทำด้วยข้อมูลที่หันกลับที่ต้องการ" ไม่ใช่ (Transformer) ซึ่งคือการหาและใช้ข้อมูลที่หันกลับ