FACULTY *of* SCIENCE *and* ENGINEERING

Department of Computer Science
and Information Systems

MIDTERM Assessment Paper

Academic Year:	2024-2025 (06/March/24)	Semester:	Spring
Module Title:	Deep Reinforcement Learning	Module Code:	CS6482
Duration of Exam:	1 Hours	Percent of Total Marks:	20
Lecturer(s):	J.J. Collins	Paper marked out of :	20

Instructions to Candidates:

- **Answer all 10 Questions.**
- All Questions are worth 2 marks each.

NAME _____

ID Number _____

S O L U T I O N S

Q1. Name an application that is typical of Symbolic/Old AI. Describe its key features. Is this application an example of Machine Learning? State the reason(s) for your answer?

Name: MYCIN, Expert System 1980

Features:

- Knowledge Base: It had a knowledge base of around 600 rules, which were derived from medical experts
- Backward Chaining: MYCIN used a backward chaining approach to infer conclusions from a set of rules
- Inference Engine: The system used an inference engine to apply these rules to the patient's data and symptoms

Is it an example Machine Learning;

No. It does not improve with Experience E at task T. Rules are fixed.

Q2: What is the formula for the update applied to a weight in a hidden layer when performing Back Propagation (BP) in a Multi-Layer Perceptron (MLP)?

For each output unit k

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

For each hidden unit h

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{h,k}\delta_k$$

Update

$$w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}$$

where

$$\Delta w_{i,j} = \eta \delta_j x_{i,j}$$

Q3: Given an grayscale input image of dimensions 100 x 100, and a convolutional layer with 5 x 5 kernels stride 1, outputting 50 feature maps of size 100 x 100. How many parameters in the convolutional layer? How many connections are there from the input layer? How many computations/operations does it perform. How much memory is required for this layer? Please show your calculations in the answer.

Number of parameters = (kh × kw × cin + 1) × cout
where:

- kh and kw are the height and width of the kernel.
- cin is the number of input channels.
- cout is the number of output channels.
- The +1 accounts for the bias term.

Parameters = ((5 x 5 x 1) + 1) x 50 = (26 * 50) = 1,300.

Each feature map contains 100 x 100 neurons =10,000 neurons
Each neuron computes a sum of (5 x 5 x 1) = 25 inputs
Total computations = 10,000 x 25 x (50) = 12.5 million computations

Memory for Parameters: The memory required for the parameters of a convolutional layer is given by:
Memoryparams = (kh×kw×cin+1) × cout × size of each parameter
where:

- kh and kw are the height and width of the kernel.
- cin is the number of input channels.
- cout is the number of output channels.
- The +1 accounts for the bias term.
- The size of each parameter depends on the data type (e.g., 4 bytes for a 32-bit float).

32 bit floating point

MemoryParams (5 x 5 x 1+1) x 50 * 32
Output = 41,600 bits = 5200 bytes

Memory for Activations: The memory required for the activations of a convolutional layer is given by:
Memory for Activations = hout × wout × cout × batch size × size of each activation
where:

- houthout and woutwout are the height and width of the output feature map.
- coutcout is the number of output channels.
- The batch size is the number of samples processed together in one forward/backward pass.
- The size of each activation depends on the data type (e.g., 4 bytes for a 32-bit float).

= (100 *100) *50 *16 * 32
= 256,000,000 bits = 32 MB

Total Memory = Memory for Parameters + Memory for Activations = 32,000,000 + 5200 = 32,005,200 bytes

Q4: What is the formula for the calculation of cross entropy? What is cross entropy? What is the approximate value for the cross entropy of containers in Figure 1?

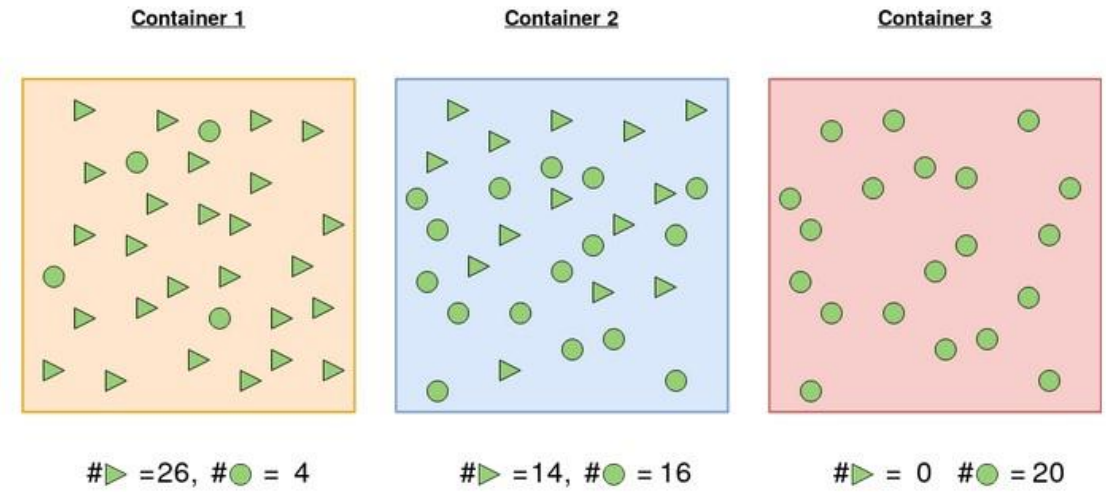


Figure 1.

Formula $H(p,q) = -\sum_{n=1} p(x)\ log(\mathrm{q(x)})$

Cross entropy is:
A measure from the field of information theory, often used in machine learning to quantify the difference between two probability distributions. Here's a breakdown:

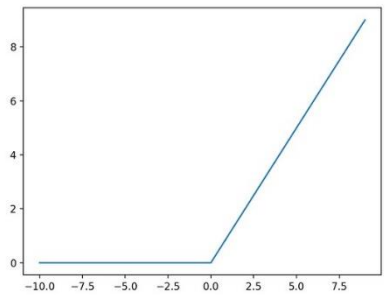
Approximate Cross Entropy → Container 1: 0.5 Container 2: 1.0 Container 3: 0

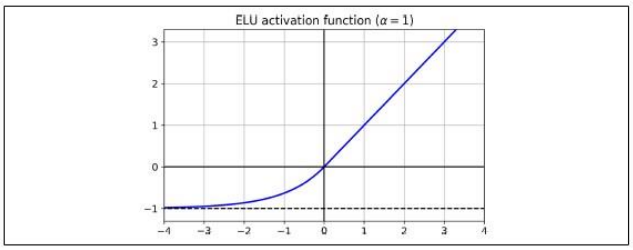
Q5: What do the acronyms RELU and ELU stand for? Draw a diagram of both. Briefly critique RELUs. Discuss why the the ELU an improvement on RELU?

RELU =

ELU = Exponential Linear Unit

Diagrams



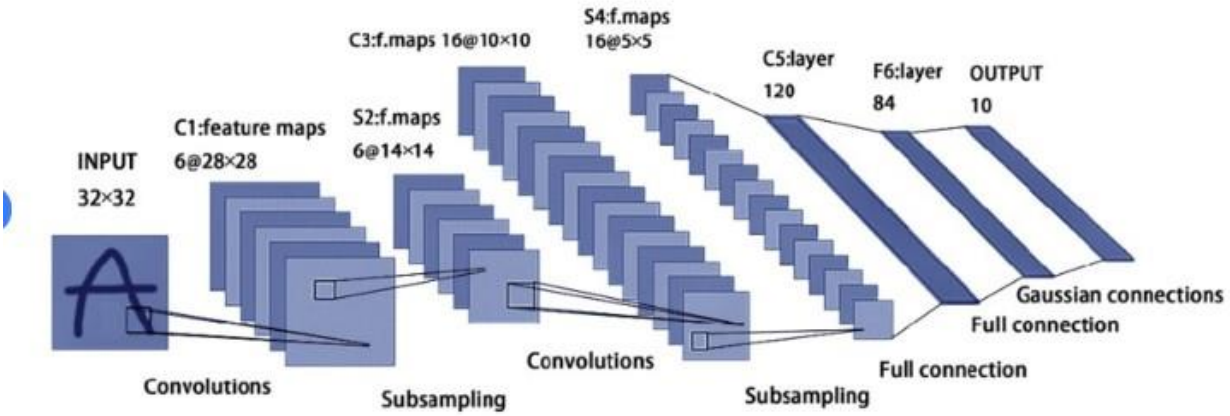


RELU ELU

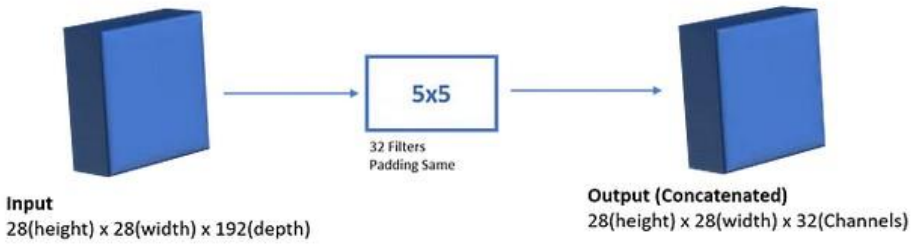
Critique of RELU. Does not provide gradient information for Z < = 0.

Why ELU is an improvement on RELU: has a slope for z <= 0 i.e. provides gradient information

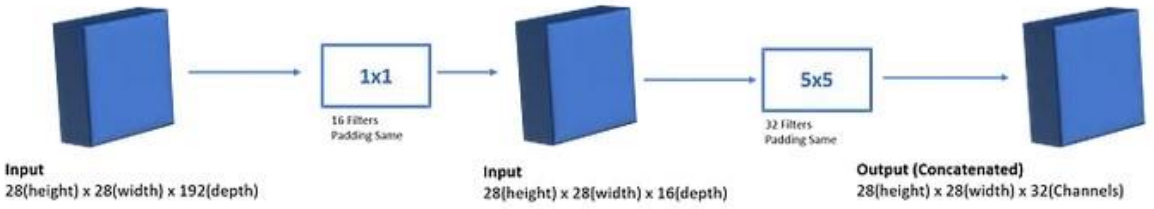
Q6. Sketch a diagram for LeNet-5 CNN. You should include input size, kernel size, and number of feature maps for the first layer.



Q7: Explain why the number of parameters in GoogleLeNet using Inception modules is significantly less than AlexNet - 6 million v of 60 million. The answer should focus exclusively on the Inception module. Illustrate the answer with a diagram and/or calculations.

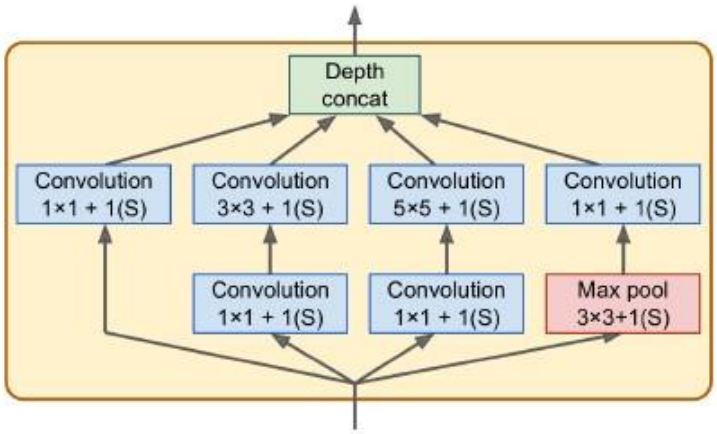


Parameters Naïve Inception = ((5 x 5 x 192) + 1) x 32 = 153,632.
Num ops: multiply the number of outputs that are required to be provided (28x28x32), with the number of multipliers needed to work out a single value within the output (5x5x192).
Num multiplier ops = (output dimensions) * (filter dimensions) * (depth of input channel)
Num multiplier ops = (28 x 28 x 32) x (5 x 5) x (192) = 120, 244, 400



Parameters Inception = (((1 x 1 x 16) + 1) x 16) + (((5 x 5 x 16) + 1) x 32) = 256 + 12,832 = 13,088

Num multiplication ops = ((28 x 28 x 16) * (1 x 1 x 192)) + ((28 x 28 x 32) * (5 x 5 x 15))
Num multiplication ops =(2,408,448)+ (10,976,000) = 12,443,648.

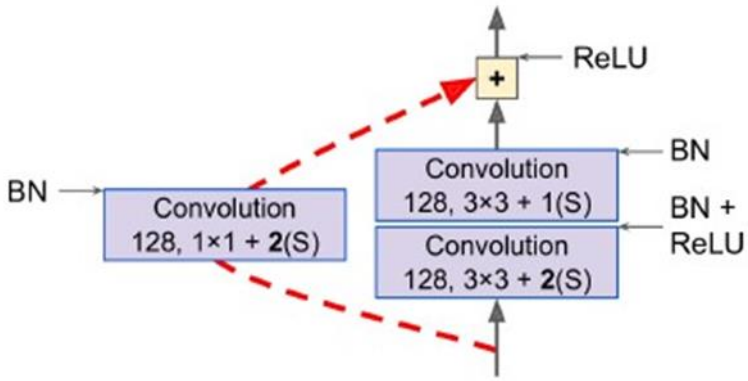


Inception Module

Q8: Describe the solution implemented in ResNet to differing input sizes when the stride changes from 1 to 2 in a convolutional layer of a residual unit.

Number of feature maps doubled every few residual units at the same time as width and height reduced
→ Inputs cannot be added to outputs

Solution: 1x1 with stride 2



Q9: Write the code to freeze the weights of the pre-trained layers in the base model when using transfer learning?

```
21. for layer in base_model.layers:
22. layer.trainable = False
```

Q10: Write the implementation for **ε**-greedy action selection.

☐ : select the best lever with probability 1-ϵ (exploit), otherwise select a random lever (explore)