**1. Physical Symbol System Hypothesis (PSSH)**

**Concept:**

* **PSSH** is a fundamental idea in artificial intelligence and cognitive science, proposed by Allen Newell and Herbert A. Simon in 1976.
* It states that a **physical symbol system** (e.g., a computer or the human brain) is both **necessary and sufficient** for general intelligence.
* A physical symbol system creates, stores, manipulates, and interprets symbols using rules and operations. These symbols represent objects, concepts, or relationships in the world.

**Example:**

* **Expert Systems** like MYCIN (used for diagnosing bacterial infections) are based on PSSH. They use symbolic reasoning and rule-based systems to mimic human decision-making.

**2. Perceptron and Activation Functions**

**Perceptron:**

* A **perceptron** is the simplest form of a neural network, consisting of:
  + **Inputs**: x1,x2,...,xn*x*1​,*x*2​,...,*xn*​
  + **Weights**: w1,w2,...,wn*w*1​,*w*2​,...,*wn*​
  + **Bias**: b*b*
  + **Activation Function**: Determines the output based on the weighted sum of inputs.

**Threshold Activation:**

* The output is 1 if the weighted sum of inputs exceeds a threshold, otherwise 0.
* Example: A perceptron with weights w1=1*w*1​=1, w2=1*w*2​=1, and bias b=−1.5*b*=−1.5 models the **AND** Boolean function.

**Update Rule for Linear Perceptron:**

* The weight update rule for a linear perceptron is:

wi←wi+η(t−o)xi*wi*​←*wi*​+*η*(*t*−*o*)*xi*​

Where:

* + wi*wi*​: weight
  + η*η*: learning rate
  + t*t*: target output
  + o*o*: actual output
  + xi*xi*​: input

**3. ReLU (Rectified Linear Unit)**

**Concept:**

* **ReLU** stands for **Rectified Linear Unit**.
* It is an activation function defined as:

f(x)=max⁡(0,x)*f*(*x*)=max(0,*x*)

* **Requirements**:
  1. **Non-linearity**: ReLU introduces non-linearity, allowing neural networks to learn complex patterns.
  2. **Sparsity**: ReLU outputs zero for negative inputs, which can lead to sparsity in activations, improving efficiency.

**4. Batch Normalization**

**Concept:**

* **Batch Normalization** is a technique used to normalize the inputs of each layer in a neural network, reducing internal covariate shift and improving training speed.
* It introduces two learnable parameters per layer: **scale (γ*γ*)** and **shift (β*β*)**.
* **Parameters**:
  + For each batch normalization layer, there are 4 parameters per neuron: γ*γ*, β*β*, mean, and variance.
  + Only γ*γ* and β*β* are trainable.

**Example Calculation:**

* In Figure 1 (from the midterm), there are 3 batch normalization layers. If each layer has 300, 100, and 10 neurons respectively:
  + Total parameters = 4×(300+100+10)=16404×(300+100+10)=1640
  + Trainable parameters = 2×(300+100+10)=8202×(300+100+10)=820

**5. Convolutional Neural Networks (CNNs)**

**Concept:**

* CNNs are used for image processing and consist of convolutional layers, pooling layers, and fully connected layers.
* **Parameters in a Convolutional Layer**:
  + Each filter has weights and a bias.
  + Number of parameters = (kernel\_size×kernel\_size×input\_channels+1)×number\_of\_filters(*kernel*\_*size*×*kernel*\_*size*×*input*\_*channels*+1)×*number*\_*of*\_*filters*

**Example Calculation:**

* For a LeNet5 CNN with input size 32x32x1, kernel size 5x5, and 6 filters:
  + Parameters per filter = 5×5×1+1=265×5×1+1=26
  + Total parameters = 26×6=15626×6=156

**6. Inception Module**

**Concept:**

* The **Inception module** is a building block in CNNs that uses multiple filter sizes (e.g., 1x1, 3x3, 5x5) in parallel to capture features at different scales.
* **Naive Inception Module**:
  + Combines outputs from multiple convolutional layers with different kernel sizes.
  + **Issue**: The naive design can lead to a large number of parameters, making the network computationally expensive.

**7. ResNet (Residual Networks)**

**Concept:**

* ResNet introduces **skip connections** (or shortcuts) that allow gradients to flow directly through the network, mitigating the vanishing gradient problem.
* **Residual Unit**:
  + Consists of a main path (convolutional layers) and a skip path (identity mapping or 1x1 convolution).
  + The output is the sum of the main path and skip path.

**Code Explanation:**

* **Lines 13-16**: If the stride is greater than 1, the skip path uses a 1x1 convolution to match dimensions.
* **Lines 30-34**: The ResNet architecture stacks multiple residual units with increasing filter sizes (64, 128, 256, 512).

**8. Reinforcement Learning (RL) Applications**

**Examples:**

1. **Healthcare**: RL is used for personalized treatment plans, optimizing drug dosages, and medical resource allocation.
2. **Finance**: RL is applied in algorithmic trading, portfolio management, and risk assessment.
3. **Energy Management**: RL is used to optimize energy consumption in smart grids and reduce costs in renewable energy systems.

**Sample Midterm Paper**

**Q1. Briefly describe the Physical Symbol System Hypothesis (PSSH). Give an example of a system that is based on PSSH. (1 mark)**

**Answer:**

* PSSH states that a physical symbol system (e.g., a computer) is necessary and sufficient for general intelligence. It manipulates symbols using rules to perform reasoning and problem-solving.
* Example: **MYCIN**, an expert system for diagnosing bacterial infections, uses symbolic reasoning and rule-based systems.

**Q2. Write the code for a perceptron that uses threshold activation and has FIXED weights w1=w2=1*w*1=*w*2=1, and bias −1.5−1.5. Name the logical Boolean function modelled by this perceptron. (1 mark)**

**Answer:**

python

Copy

def perceptron(x1, x2):

w1, w2 = 1, 1

bias = -1.5

weighted\_sum = w1 \* x1 + w2 \* x2 + bias

return 1 if weighted\_sum > 0 else 0

* This perceptron models the **AND** Boolean function.

**Q3. What is the update rule for a linear perceptron? (1 mark)**

**Answer:**

* The update rule is:

wi←wi+η(t−o)xi*wi*​←*wi*​+*η*(*t*−*o*)*xi*​

Where:

* + wi*wi*​: weight
  + η*η*: learning rate
  + t*t*: target output
  + o*o*: actual output
  + xi*xi*​: input

**Q4. Describe the 2 requirements that underpin RELUs. What does the acronym RELU stand for? (1 mark)**

**Answer:**

* **ReLU** stands for **Rectified Linear Unit**.
* **Requirements**:
  1. **Non-linearity**: ReLU introduces non-linearity, allowing neural networks to learn complex patterns.
  2. **Sparsity**: ReLU outputs zero for negative inputs, leading to sparsity in activations.

**Q5. How many parameters are in the three Batch Normalization layers in Figure 1? Of these, how many are trainable? Show the calculations. (1 mark)**

**Answer:**

* Total parameters = 4×(300+100+10)=16404×(300+100+10)=1640
* Trainable parameters = 2×(300+100+10)=8202×(300+100+10)=820

**Q6. How many parameters are in the first convolutional layer of a LeNet5 CNN where the input is 32x32x1, the kernel is 5x5, and there are six such filters? Show the calculations. (2 marks)**

**Answer:**

* Parameters per filter = 5×5×1+1=265×5×1+1=26
* Total parameters = 26×6=15626×6=156

**Q7. Draw a diagram for a Naive Inception module. What is the issue with this design? (2 marks)**

**Answer:**

* **Diagram**: A naive Inception module combines 1x1, 3x3, and 5x5 convolutions in parallel.
* **Issue**: The naive design leads to a large number of parameters, making the network computationally expensive.

**Q8. Explain the intent/purpose of the code in (a) lines 13-16, and (b) lines 30-34 in the ResNet architecture. (2 marks)**

**Answer:**

* **(a) Lines 13-16**: If the stride is greater than 1, the skip path uses a 1x1 convolution to match dimensions.
* **(b) Lines 30-34**: The ResNet architecture stacks multiple residual units with increasing filter sizes (64, 128, 256, 512).

**Q9. Give three good examples of the use of Reinforcement Learning that is different from those used in lectures. (2 marks)**

**Answer:**

1. **Healthcare**: Personalized treatment plans and drug dosage optimization.
2. **Finance**: Algorithmic trading and portfolio management.
3. **Energy Management**: Optimizing energy consumption in smart grids.