Keerthi Vasan A
212222240048

6B) In a practical DL project, how do you determine the most suitable function for different layers in a neural network and why? Additionally, explain these activation functions and provide PyTorch implementations?

7A) Explain any 3 CNN architecture, discussing their key components, advantages and applications.

8B) VGG16, CIFAR-10 pytorch.

i) Load the CIFAR dataset.

ii) Use a pre-trained VGG16 model as the base model.

iii) Create a Sequential model with the appropriate number of neurons in the output layer, activation function, and loss function.

iv) Train the model with training data and validation data.

6B) Activation Functions:-

* Activation function is a function that is implemented in the hidden layers of the neuron that is activated when the signals are transmitted from the input layer.

* These activation functions improves the effect of each input signals so that the loss function is reduced and the accuracy is improved.

* There are different types of activation function which are used based on the requirements. they are:

* Sigmoid.
* Tanh
* ReLU.
* Leaking ReLU
* PReLU
* ELU
* Softmax.

Sigmoid:

* Sigmoid activation function is used for the Binary classification process.

* The range of the output will be

SL.NO.:

ranging from 0 to 1.

Implementation:

```
sigmoid = nn.Sigmoid(torch.tensor(2.0, 1.0, 0.5))
output = sigmoid.model()
```

Tanh:

* Tanh is also called as Hyperbolic Tangent activation function.

* The output of Tanh ranges from -1 to 1.

Implementation:

```
tanh = nn.TanH(torch.tensor(2.0, 1.0, 0.5))
output = tanh.model()
```

ReLU:

* ReLU (Rectified Linear Unit). ReLU focusses on Zero-centered outputs.

* ReLU activation function ranges from 0 to infinity as it adds the signals

as lists

```
relu = nn.ReLU(torch(2.0, 1, 0.5))
output = relu..model()
```

Leaking ReLU:-

* Leaky ReLU or Leaking ReLU can be used to avoid the dying neuron problem and fixing it.

Implementation:

```
lrelu = nn.LReLU(torch(2.0, 1, 0.5))
output = lrelu.model().
```

PReLu:

* PReLU is the short form of Parametric Rectified Linear Unit.

* PReLU is a very good activation function to avoid the vanishing gradient problem.

```
prelu = nn.PReLU(torch(2.0, 1.0, 0.5))
output = prelu.model().
```

ELU:

* ELU is the short form of Exponential Linear Unit.

* The ELU activation function performs well even in critical conditions like vanishing gradient problem and dying neurons.

```
elu = nn.ELU(torch(0,1,0.5))
output = elu.model().
```

Softmax:

* Softmax is one among the well known activation function and mostly implemented for smaller threads.

* Softmax is also good in the classification process and can expect accurate results.

7A) CNN Architectures:

* CNN stands for Convolutional Neural Networks which has convolutional layers with different ranges is present.

* There are many types of CNN architectures, such as:

* LeNet - 98
* AlexNet
* ResNet.

i) LeNet:

* The LeNet architecture is the oldest architecture of the convolution Neural Network.

* It was invented in the year of 1998.

* LeNet has 2 Convolution layers and 2 fully connected layers.

* LeNet uses softmax activation function and can classify zero to nine input signals for a single loop.

* The inputs are 32 x 32 grayscale images

Advantages of LeNet:

* LeNet architecture and perform well for small number of inputs and the accuracy will be clear.

* It was the fastest known architecture in its time.

Applications:

* It is applied in character recognition from various input files like images and PDFs.

* OCR (Optical Character Recognition) works in LeNet architecture.

ii) AlexNet :

* Alex Net is invented in the year of 2012.

* Alex Net architecture can have five convolutional layers and three fully connected layers.

* The input can 227 x 227 gray scale image.

* Alex Net uses ReLU for processing and Softmax for output generation.

* It can classify around 1000 images for a single loop.

Advantages:

* Can be used in the image classification method. for higher pixel images.

* Faster and easy implementation can be seen.

SAVEETHA ENGINEERING COLLEGE
AUTONOMOUS
NAAC A GRADE
Affiliated to Anna University | Approved by AICTE

Sl. No.:

Marks

_____ Examination –April/May/Nov/Dec 20___

Register No.:

Branch & Sem.:

Date:

Sign of the Invigilator:

Subject Code & Subject Name :

Application:

* Alex Net is implemented in the ImageNet.

* It can be also used for digital image processing, in medical images,

iii) ResNet:

* The ResNet architecture is invented in the year of 2015.

* The input image ranges from 30 to 50 and 100.

* ResNet uses softmax activation function.

* ResNet consists of convolutional network and can have any number of fully connected layers based on the requirements.

Advantages and Implementation:

* ResNet gives faster results and the desired results are very accurate.

* It is implemented in Medical image processing and Facial recognition.

8b) i) Loading CIFAR-10 Dataset :-

```
import torch
from torch import torchvision as nn.

torch = torchvision ( torchvision. Reshape (224),
                      torchvision. Replace = True).

train = torch. dataset. CIFAR10 ( train = True, download
                                  = True, shuffle = True).

train_loader = torchvision. Dataloader (train).

test = torch. dataset. CIFAR10 ( train = False,
                                 download = True, shuffle = True).

test_loader = torchvision. Dataloader (test).
```

ii) Using "VGG16 model as base" model:-

```
from torchvision import models
vgg = models.vgg16().
for vgg in parameters():
    vgg = param.requires_grad().
```

iii) Creating Sequential Model:-

```
criterion = torch.CrossEntropyLoss().
```

iv) Training

```
epochs = 5.
for vgg in range():
    running_loss = 0.0
    for param in parameters:
        loss = criterion.loss()
```

```
running_loss += loss.

print("Loss =", running_loss / len(train)
```