**Reid Jackson**

**4442 Assignment 4**

**April 2nd 2020**

**250914839**

**2)**

**a)**

Pseudo-code for Forward Propagation

```
Function MaxPool(x, poolWidth, poolHeight, stride)

N: Batch size
x: input signal
H: Image Height (x.height)
W: Image Width (x.width)
H’: Pooled Image Height (from pool
W’: Pooled Image Width
S: stride (1 in this case)
HP: Pool Height
WP: Pool Weight

HO (Height of Output) = 1 + (H - HP) / S
WO (Height of Width) = 1 + (W - WP) / S

Output = array of size HO x WO

For r in range (0, H, S) // Slide vertically from 0 to H with stride S

    For c in range (0, W, S) // Slide horizontally from 0 to W with stride S

        Get the largest element in the filter

        Put the largest element in Output at position [r/S] [c/S]
        // so first pool will be 0,0 then 0,1…  1,0 then 1,1 …  h,w

cache = a tuple (x, poolWidth, poolHeight, stride) // Where this saves a cache for back propogation

return Output, cache
```

Pseudo-code for Backward Propagation

```
Function MaxPoolBackward(dout, cache)

x, poolWidth, poolHeight, stride = cache (tuple returned from forward propogation)

dout: derivatives

H = x.height
W = x.width

dx = an array for zeros, same shape as x, so H x W // derivatives for the backward values to go

For r in range (0, H) // Slide vertically from 0 to H
```

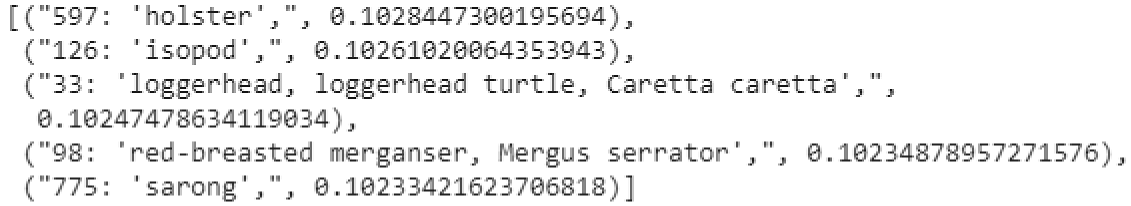
```
    For c in range (0, W) // Slide horizontally from 0 to W
        Take the pool value from the pool layer, from the x cache determine what value was picked
        For the value picked, dx at that position gets set to the value of dout. Meaning that the max pooled value picked is set in the ‘original’ position (rest are 0s from intitialization)
return dx
```

**b)**

For boundaries I pad the input volume with zeros around the border. Zero padding allows for the control of the spatial size of the output volumes. So that if the stride doesn’t properly work for the input size ex 7x7 input, 2x2 filter and a stride of 2 will overlap the right and bottom, so padding the input with 0s will allow for the output to of integer dimensions.

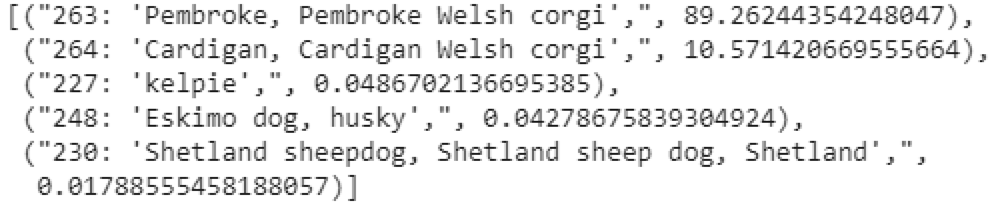
**3)**

- **a)** There are 4096 features for the input the last layer for AlexNet.
- **b)** The top five predictions without pretraining are random every time it is run, here is an example of the output, with the last float number value being the percent confidence.



```
[("597: 'holster',", 0.1028447300195694),
 ("126: 'isopod',", 0.10261020064353943),
 ("33: 'loggerhead, loggerhead turtle, Caretta caretta',",
  0.10247478634119034),
 ("98: 'red-breasted merganser, Mergus serrator',", 0.10234878957271576),
 ("775: 'sarong',", 0.10233421623706818)]
```

- **c)** Running it again with pretraining to be true here are the five predicitons with the last float number value being the percent confidence.



```
[("263: 'Pembroke, Pembroke Welsh corgi',", 89.26244354248047),
 ("264: 'Cardigan, Cardigan Welsh corgi',", 10.571420669555664),
 ("227: 'kelpie',", 0.0486702136695385),
 ("248: 'Eskimo dog, husky',", 0.04278675839304924),
 ("230: 'Shetland sheepdog, Shetland sheep dog, Shetland',",
  0.01788555458188057)]
```

So a Pembroke Welsh Corgi is the top prediction with 89.26% confidence!