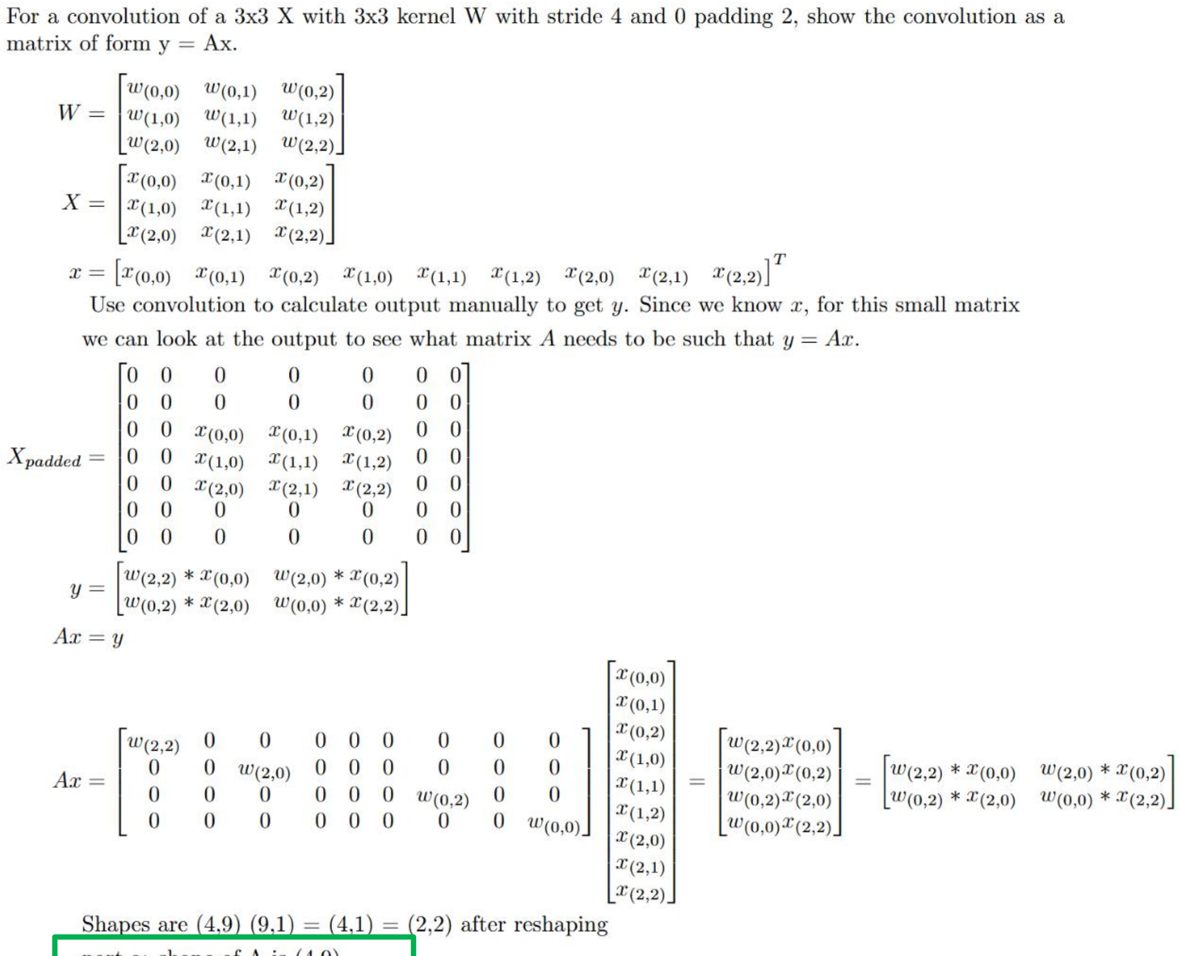
Assignment 2 Theory Problem Set

DO NOT TAG

Name: Julie Cha
GT Email:jcha73@gatech.edu

Theory PS Q1. Must show your work for full credit. Feel free to add extra slides if needed.



For a convolution of a 3x3 X with 3x3 kernel W with stride 4 and 0 padding 2, show the convolution as a matrix of form y = Ax.

$$W = \begin{bmatrix} w_{(0,0)} & w_{(0,1)} & w_{(0,2)} \\ w_{(1,0)} & w_{(1,1)} & w_{(1,2)} \\ w_{(2,0)} & w_{(2,1)} & w_{(2,2)} \end{bmatrix}$$

$$X = \begin{bmatrix} x_{(0,0)} & x_{(0,1)} & x_{(0,2)} \\ x_{(1,0)} & x_{(1,1)} & x_{(1,2)} \\ x_{(2,0)} & x_{(2,1)} & x_{(2,2)} \end{bmatrix}$$

$$x = \begin{bmatrix} x_{(0,0)} & x_{(0,1)} & x_{(0,2)} & x_{(1,0)} & x_{(1,1)} & x_{(1,2)} & x_{(2,0)} & x_{(2,1)} & x_{(2,2)} \end{bmatrix}^T$$

Use convolution to calculate output manually to get y. Since we know x, for this small matrix we can look at the output to see what matrix A needs to be such that $y = Ax$.

$$X_{padded} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_{(0,0)} & x_{(0,1)} & x_{(0,2)} & 0 & 0 \\ 0 & 0 & x_{(1,0)} & x_{(1,1)} & x_{(1,2)} & 0 & 0 \\ 0 & 0 & x_{(2,0)} & x_{(2,1)} & x_{(2,2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$y = \begin{bmatrix} w_{(2,2)} * x_{(0,0)} & w_{(2,0)} * x_{(0,2)} \\ w_{(0,2)} * x_{(2,0)} & w_{(0,0)} * x_{(2,2)} \end{bmatrix}$$

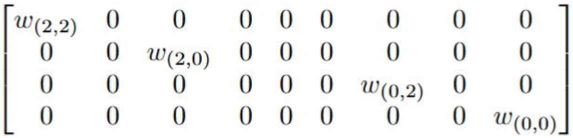
$$Ax = y$$

$$Ax = \begin{bmatrix} w_{(2,2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_{(2,0)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w_{(0,2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{(0,0)} \end{bmatrix} \begin{bmatrix} x_{(0,0)} \\ x_{(0,1)} \\ x_{(0,2)} \\ x_{(1,0)} \\ x_{(1,1)} \\ x_{(1,2)} \\ x_{(2,0)} \\ x_{(2,1)} \\ x_{(2,2)} \end{bmatrix} = \begin{bmatrix} w_{(2,2)}x_{(0,0)} \\ w_{(2,0)}x_{(0,2)} \\ w_{(0,2)}x_{(2,0)} \\ w_{(0,0)}x_{(2,2)} \end{bmatrix} = \begin{bmatrix} w_{(2,2)} * x_{(0,0)} & w_{(2,0)} * x_{(0,2)} \\ w_{(0,2)} * x_{(2,0)} & w_{(0,0)} * x_{(2,2)} \end{bmatrix}$$

Shapes are (4,9) (9,1) = (4,1) = (2,2) after reshaping

part a: shape of A is (4,9)

Part b: entries of matrix A



$$\begin{bmatrix} w_{(2,2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_{(2,0)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w_{(0,2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{(0,0)} \end{bmatrix}$$

Theory PS Q2. Must show your work for full credit. Feel free to add extra slides if needed.

Please write a box around your answers.

Calculate W, b for $x_0 = 2$, $x_0 = -1$, $x_0 = 1$ for this relu network.

$$h(x) = W^{(3)} \max\left\{0,\ W^{(2)} \max\left\{0,\ W^{(1)}x + b^{(1)}\right\} + b^{(2)}\right\} + b^{(3)}$$

$$\begin{aligned}
z_1 &= W_1 x + b_1 \\
a_1 &= \mathrm{ReLU}(z_1) \\
z_2 &= W_2 a_1 + b_2 \\
a_2 &= \mathrm{ReLU}(z_2) \\
z_3 &= W_3 a_2 + b_3
\end{aligned}$$

Calculate $h(x_0)$ for each value of x_0

Get $\frac{\partial h(x_0)}{\partial x}$ for each value of x_0

Get ReLU and derivative of ReLU element wise, in our case we have 2x2 matrices. Return deriv matrix with calculated values along its diagonal

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

$$derivReLU(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

$$derivReLU(z) = diag(derivReLU(z))$$

$$\frac{\partial h(x)}{\partial x} = \frac{\partial z_3}{\partial a_2} \frac{\partial a_2}{\partial z_2} \frac{\partial z_2}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial x}$$

$$\frac{\partial z_3}{\partial a_2} = W_3$$

$$\frac{\partial a_2}{\partial z_2} = derivReLU(z_2)$$

$$\frac{\partial z_2}{\partial a_1} = W_2$$

$$\frac{\partial a_1}{\partial z_1} = derivReLU(z_1)$$

$$\frac{\partial z_1}{\partial x} = W_1$$

$$\frac{\partial h(x)}{\partial x} = \frac{\partial z_3}{\partial a_2} @ \frac{\partial a_2}{\partial z_2} @ \frac{\partial z_2}{\partial a_1} @ \frac{\partial a_1}{\partial z_1} @ \frac{\partial z_1}{\partial x}$$

$$\frac{\partial h(x)}{\partial x} = W_3 @ derivReLU(z_2) @ W_2 @ derivReLU(z_1) @ W_1$$

Calculate $W = \frac{\partial h(x)}{\partial x}$

Since $Wx_0 + b = h(x_0)$ then

Calculate $b = h(x_0) - Wx_0$

$$W_1 = \begin{bmatrix}1.5\\0.5\end{bmatrix}$$

$$b_1 = \begin{bmatrix}0\\1\end{bmatrix}$$

$$W_2 = \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}$$

$$b_2 = \begin{bmatrix}0\\1\end{bmatrix}$$

$$W_3 = \begin{bmatrix}1 & 1\end{bmatrix}$$

$$b_3 = -1$$

$$h(x) = \begin{bmatrix}1 & 1\end{bmatrix} \max\left\{0, \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix} \max\left\{0, \begin{bmatrix}1.5\\0.5\end{bmatrix} x + \begin{bmatrix}0\\1\end{bmatrix}\right\} + \begin{bmatrix}0\\1\end{bmatrix}\right\} +$$

For $x_0 = 2$ calculate $h(x_0)$

$$z_1 = W_1 x + b_1 = \begin{bmatrix}1.5\\0.5\end{bmatrix} 2 + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}3\\2\end{bmatrix}$$

$$a_1 = \text{ReLU}(z_1) = \begin{bmatrix}3\\2\end{bmatrix}$$

$$z_2 = W_2 a_1 + b_2 = \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix} \begin{bmatrix}3\\2\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}7\\9\end{bmatrix}$$

$$a_2 = \text{ReLU}(z_2) = \begin{bmatrix}7\\9\end{bmatrix}$$

$$h(2) = z_3 = W_3 a_2 + b_3 = \begin{bmatrix}1 & 1\end{bmatrix} \begin{bmatrix}7\\9\end{bmatrix} + (-1) = 15$$

For $x_0 = 2$ calculate $\frac{\partial h(x)}{\partial x}$ at x_0

$$\frac{\partial z_3}{\partial a_2} = W_3 = \begin{bmatrix}1 & 1\end{bmatrix}$$

$$\frac{\partial a_2}{\partial z_2} = derivReLU(z_2) = derivReLU(\begin{bmatrix}7\\9\end{bmatrix}) = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

$$\frac{\partial z_2}{\partial a_1} = W_2 = \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}$$

$$\frac{\partial a_1}{\partial z_1} = derivReLU(z_1) = derivReLU(\begin{bmatrix}3\\2\end{bmatrix}) == \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

$$\frac{\partial z_1}{\partial x} = W_1 = \begin{bmatrix}1.5\\0.5\end{bmatrix}$$

$$\frac{\partial h}{\partial x} = \begin{bmatrix}1 & 1\end{bmatrix} \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix} \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \begin{bmatrix}1.5\\0.5\end{bmatrix}$$

$$\frac{\partial h}{\partial x} = \begin{bmatrix}1 & 1\end{bmatrix} \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix} \begin{bmatrix}1.5\\0.5\end{bmatrix}$$

$$\frac{\partial h}{\partial x} = 6$$

For $x_0 = 2$ calculate W, b from $h(x_0)$, $\frac{\partial h(x)}{\partial x}$ at x_0

$$W = \frac{\partial h(x)}{\partial x} = 6$$

$$b = h(x_0) - W x_0 = 15 - 6(2) = 3$$

For $x_0 = 1$ calculate $h(x_0)$

$$
\begin{aligned}
z_1 &= W_1 x + b_1 = \begin{bmatrix}1.5\\0.5\end{bmatrix} 1 + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}1.5\\0.5\end{bmatrix}\\
a_1 &= \text{ReLU}(z_1) = \begin{bmatrix}1.5\\0.5\end{bmatrix}\\
z_2 &= W_2 a_1 + b_2 = \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}1.5\\0.5\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}4.5\\5.5\end{bmatrix}\\
a_2 &= \text{ReLU}(z_2) = \begin{bmatrix}4.5\\5.5\end{bmatrix}\\
h(2) = z_3 &= W_3 a_2 + b_3 = \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}4.5\\5.5\end{bmatrix} + (-1) = 9
\end{aligned}
$$

For $x_0 = 1$ calculate $\frac{\partial h(x)}{\partial x}$ at x_0

$$
\begin{aligned}
\frac{\partial z_3}{\partial a_2} &= W_3 = \begin{bmatrix}1 & 1\end{bmatrix}\\
\frac{\partial a_2}{\partial z_2} &= derivReLU(z_2) = derivReLU(\begin{bmatrix}4.5\\5.5\end{bmatrix}) = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\\
\frac{\partial z_2}{\partial a_1} &= W_2 = \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\\
\frac{\partial a_1}{\partial z_1} &= derivReLU(z_1) = derivReLU(\begin{bmatrix}1.5\\0.5\end{bmatrix}) == \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\\
\frac{\partial z_1}{\partial x} &= W_1 = \begin{bmatrix}1.5\\0.5\end{bmatrix}\\
\frac{\partial h}{\partial x} &= \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1.5\\0.5\end{bmatrix}\\
\frac{\partial h}{\partial x} &= \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}1.5\\0.5\end{bmatrix}\\
\frac{\partial h}{\partial x} &= 6
\end{aligned}
$$

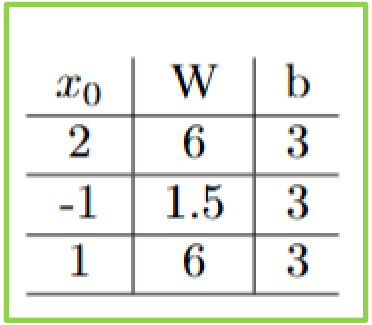
For $x_0 = 1$ calculate W, b from $h(x_0)$, $\frac{\partial h(x)}{\partial x}$ at x_0

$$
\begin{aligned}
W &= \frac{\partial h(x)}{\partial x} = 6\\
b &= h(x_0) - W x_0 = 9 - 6(1) = 3
\end{aligned}
$$

For $x_0 = -1$ calculate $h(x_0)$

$$
\begin{aligned}
z_1 &= W_1 x + b_1 = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} (-1) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1.5 \\ 0.5 \end{bmatrix} \\
a_1 &= \mathrm{ReLU}(z_1) = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \\
z_2 &= W_2 a_1 + b_2 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} \\
a_2 &= \mathrm{ReLU}(z_2) = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} \\
h(2) = z_3 &= W_3 a_2 + b_3 = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} + (-1) = 1.5
\end{aligned}
$$

For $x_0 = -1$ calculate $\frac{\partial h(x)}{\partial x}$ at x_0



x_0	W	b
2	6	3
-1	1.5	3
1	6	3

$$
\begin{aligned}
\frac{\partial z_3}{\partial a_2} &= W_3 = \begin{bmatrix} 1 & 1 \end{bmatrix} \\
\frac{\partial a_2}{\partial z_2} &= derivReLU(z_2) = derivReLU(\begin{bmatrix} 1 \\ 1.5 \end{bmatrix}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
\frac{\partial z_2}{\partial a_1} &= W_2 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \\
\frac{\partial a_1}{\partial z_1} &= derivReLU(z_1) = derivReLU(\begin{bmatrix} -1.5 \\ 0.5 \end{bmatrix}) == \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\
\frac{\partial z_1}{\partial x} &= W_1 = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} \\
\frac{\partial h}{\partial x} &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} \\
\frac{\partial h}{\partial x} &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} \\
\frac{\partial h}{\partial x} &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \\
\frac{\partial h}{\partial x} &= 1.5
\end{aligned}
$$

For $x_0 = -1$ calculate W, b from $h(x_0)$, $\frac{\partial h(x)}{\partial x}$ at x_0

$$
\begin{aligned}
W &= \frac{\partial h(x)}{\partial x} = 1.5 \\
b &= h(x_0) - W x_0 = 1.5 - 1.5(-1) = 3
\end{aligned}
$$

Theory PS Q3. Keep your answer concise yet complete.

In practice, the ReLU function dead neurons problem of producing 0 gradients when the activation is less than 0 is not a big problem. In theory, once the activation function outputs zero, the neuron would always output zero and not contribute to learning.

In practice, factors avoiding reducing the risk of a dead neuron problem include:
-random initialization or Xavier initialization, hence weight initialization unlikely to cause large numbers of ReLUs to produce negative or zero outputs.
- not using large negative biases that would cause the neuron output to be less than or equal to zero
- batch normalization, renormalizes the the activations within a layer so that its unlikely that large numbers of neurons are inactive
- single step in SGD even in mini-batch has multiple data points, so unlikely to have all slopes zero with no learning
- avoidance of excessively large learning rates that can cause gradients to vanish with inactive neurons

Assignment 2 Paper Review

DO NOT TAG

Provide a short preview of the paper of your choice.

The main contribution of this paper is showing that ImageNet trained CNNs are strongly biased towards making classifications based on recognizing textures instead of shapes. From human behavioral science we know that people look more strongly at shapes and outlines rather than textures when making image classification decisions. The key insight is challenging the traditionally held assumptions that CNNs in object recognition tasks use object shape most strongly, the paper calls this the shape hypothesis. This paper provided strong evidence for a texture hypothesis, which is that CNNs trained in ImageNet tend to rely more heavily on just recognizing the textures for each classification. A possible explanation is that information from local patches alone is sufficient for excellent performance. The strength of this paper is providing excellent support for the texture hypothesis by showing that model performance reduced dramatically on ImageNet variants without textural information – black silhouettes, images with edges only, or images with texture from another classification. Training a model on a Stylized-ImageNet (SIN) with randomly selected textures applied to images was able to overcome the texture bias of CNNs and now have it focus more on the shapes of the images. A weakness of the paper could be not showing this same result on another dataset besides ImageNet, but it is something that would be great for future steps.

I was blown away by this paper since I had no idea that CNNs were biased toward textures instead of object shapes, which logically is so different from now humans process images. I was also impressed that in addition the paper showed that with training, the bias could be changed toward object shape recognition and that this had multiple benefits in model performance in generalizability. In applications in which shape based recognition is important, perhaps model pretraining to modify biases would be helpful for the project.

Paper specific Q1. Feel free to add extra slides if needed.

It is important to understand the biases of the neural network, even if it is performing well on the training set. With AI models, explainability and methodology used in addition to purely performance are important. The paper shows that ImageNet trained CNNs have a texture bias, which is contrast to humans which have a shape bias in object recognition. Understanding how a pretrained ImageNet works is important when trying to apply it to different datasets with transfer learning. It now makes sense why the pretrained model may perform very poorly on datasets with images with just silhouettes, edges, or lack of textural cues that correctly align with the classification. I think that it is ideal for a model to have the same biases as humans, since we often use domain knowledge to help design experiments and the image augmentations for the experiment. The domain knowledge may be less helpful if the model is not understanding or utilizing the curated features in a similar way as the domain expert. In addition, shape based bias similar to humans has additional benefits such as better robustness against distortions even those it had never seen in training and better transfer learning in object detection.

Training on stylized images changes the biases of the network to shape based, because it is no longer able to make correct classifications solely by looking at local object related textural information. Therefore, the model is forced to learn by integrating and classifying global shapes of the object. This bias generalizes better to datasets with corruptions because it can still recognize the object shape information which is often still preserved while corruptions may degrade textural information through distortions like phase noise, contrast changes, or high/low pass filtering.

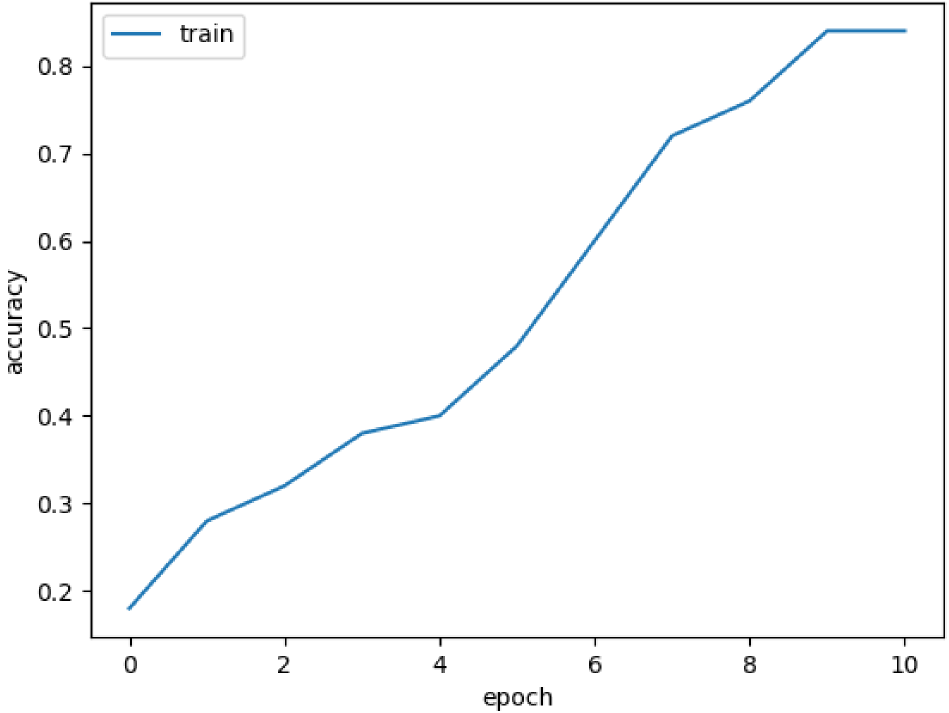
Assignment 2 Writeup

DO NOT TAG

Part-1 ConvNet

DO NOT TAG

Put your training curve here:



My CNN Model

DO NOT TAG

Describe and justify your model design in plain text here:

My model for CIFAR10 consisted of 5 conv filters followed by 3 fully connected layers, with max pooling at some intervals and ReLU after each convolution kernel. The first conv was 7x7 with stride 1, padding 2, and 96 filters filters, followed by max pooling with kernel size 2 and stride 2. The second conv was 5x5 with stride 1, padding 1, and 96 filters.The third conv was 5x5 with stride 1, padding same, and 128 filters, followed by max pooling with kernel size 2 and stride 2. The fourth conv was 5x5 with stride 1, padding same, and 164 filters. The fifth conv was 3x3 with stride 1, padding 0, and 196 filters followed by 0.25 dropout.

The three fully connected layers following have input and output features numbers as follows: 3136> 500 -> 80 -> 10 prior to input into softmax for classification. Dropout of 0.25 after the first layer.

My model design consisted of adding additional convolutional layers, since a two layer CNN was found to be underfitting the data and additional model complexity was required. The multiple convolutions allow efficiently extracting features from the images with fewer parameters than with a pure neural network for efficiency in model performance vs size. The filter size decreases along with an increase in the number of filters, as we try to consolidate the information from the larger original image. The three fully connected layers at the end gather information all the different kernels for global information prior to producing features for input into softmax for classification. Dropout is applied to reduce overfitting , so that the model does not memorize the training set.

Describe and justify your choice of hyper-parameters:

batch size: 96, learning rate: 0.01, reg = 0.001, epochs=20, momentum=0.95,dropout=0.25,
steps = [12,14], warmup:0, gamma=1, loss_type=CE

A smaller batch size was used to add a small regularization effect since the values calculated are a bit noisier, the learning rate was increased for faster training with no divergence issues noted, the regularization was increased by a small amount and dropout was also introduced to help reduce overfitting, slight increase in momentum to encourage faster optimization, increase in number of epochs from 10 to 20 since the loss was still decreasing. Steps numbers increased so that reduce in learning rate occurs at steps closer to the end of the 20 epoch run as the local minimum is reached.

What’s your final accuracy on validation set?

0 .8470

Data Wrangling

DO NOT TAG

What’s your result of training with regular CE loss on imbalanced CIFAR-10?

Accuracy: 0.6889

Tune appropriate parameters and fill in your best per-class accuracy in the table

CE loss Model	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Accuracy per Class	0.89	0.98	0.75	0.63	0.73	0.53	0.84	0.65	0.40	0.48

What’s your result of training with CB-Focal loss on imbalanced CIFAR-10?
Accuracy: 0.6493

Additionally tune the hyper-parameter beta and fill in your per-class accuracy in the table; add more rows as needed

Best focal loss model	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Beta = 0.999	0.91	0.93	0.60	0.56	0.58	0.55	0.71	0.65	0.50	0.51

	Class 0	Class1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
beta=0.9	0.94	0.98	0.76	0.69	0.72	0.51	0.62	0.51	0.36	0.24
beta=0.99	0.96	0.98	0.77	0.71	0.75	0.53	0.65	0.54	0.46	0.44

Put your results of CE loss and CB-Focal Loss(best) together:

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
CE Loss	0.89	0.98	0.75	0.63	0.73	0.53	0.84	0.65	0.40	0.48
CB-Focal	0.91	0.93	0.60	0.56	0.58	0.55	0.71	0.65	0.50	0.51

Describe and explain your observation on the result:

*Explanation should go into **WHY** things work the way they do in the context of Machine Learning theory/intuition, along with justification for your experimentation methodology. **DO NOT** just describe the results, you should explain the reasoning behind your choices and what behavior you expected. Also, be cognizant of the best way to mindful and show the results that best emphasizes your key observations. If you need more than one slide to answer the question, you are free to create new slides.*

Gamma = 0 would make focal loss inactive and equivalent to cross entropy loss, while higher values of gamma emphasize accuracy on harder to classify classes. Overly high values of gamma caused a marked reduction in overall accuracy. The misbalance in the CE resnet-32 is between 0.4 to 0.9, so a gamma value of 2 was utilized. This provides solid reduction in loss for easy examples, but had only a moderate increase in balance of class accuracies with the lowest class accuracies moved up to 0.5.

Tuning the beta values changes the class balanced loss, with beta = 0 defaulting to no reweighting. High beta values like 0.999 emphasize rare classes, lower values of beta do less reweighting towards rare classes. Higher values of beta caused a marked reduction in overall accuracy. My best focal loss used a beta value of 0.999. A lower value of beta like 0.9 or 0.99 had outcomes that were closer to the CE model in the aspect that the distribution of class accuracies was very skewed. while a higher beta value like 0.999 had slighltly better evenness of class accuracies but a lower overall accuracy.

The regularization was changed from 0.0005 for CE to 0.005 for focal loss. The reason is that more heavy regularization was used to prevent the focal loss model from memorizing training examples of rare/hard to classify examples.

The number of epochs was increased to 50 epochs, to allow the focal loss model more time to converge along with step lr reduction at 40,45 epochs as the model got closer to the local minimum.
The learning lesson from this example shows that focal loss models are very difficult to tune compared to CE, and accuracy often suffers. The learning rate was also increased to 0.01, to improve convergence speed since no divergence or oscillations were found with this increase in learning rate.

I selected beta = 0.999 as my best focal loss model even though its overall accuracy at 0.6493 was less than CE at 0.6889 and even another focal loss model with lower beta at 0.99 at accuracy = 0.6778. The rationale is that beta = 0.999 had a slight improvement of class accuracies of the lowest performing classes (#8,9), which was not possible when using only a CE only model. I went with the assumption that this was the priority since FL was requested.

Ideally to maximize performance, the next step would have been to used CE and focal loss models in combination, either in sequence or as a hybrid such as a basic example of Loss = 1*CE + (1-alpha) FL. This would combine the more stable optimization of CE for a higher overall accuracy during bulk training along with the benefits of FL to improve performance on rare and difficult to classify classes.

```
CE model--------------
Train:
  batch_size: 128
  learning_rate: 0.1
  reg: 0.0005
  epochs: 50
  steps: [100, 101]
  warmup: 0
  momentum: 0.9
  gamma: 1

network:
  model: ResNet-32
  save_best: True

data:
  imbalance: imbalance
beta: 0.9999

loss:
  loss_type: CE
```

```
Best Focal Loss Model--------------------------------
Train:
  batch_size: 128
  learning_rate: 0.01
  reg: 0.005
  epochs: 50
  steps: [40, 45]
  warmup: 0
  momentum: 0.9
  gamma: 2

network:
  model: ResNet-32
  save_best: True

data:
  imbalance: imbalance
beta: 0.999

loss:
  loss_type: Focal
```