Special Topics: Deep Learning

Sheet 1 — COMP 499/691

Assignment 2

Due Date: April 6

Submission: Submit a pdf of your written answers and code and output from programming portions. You will also submit a runnable ipynb (single file) including all answers to programming portions of the questions.

Note: The assignments are to be done individually. You may discuss ideas regarding questions with others but should not share answers. List the names of anyone you have extensively discussed a question with at the top of your final submission.

Eugene Belilovsky: eugene.belilovsky@concordia.ca

1. (a) **(10 points)** Consider the AlexNet Model designed for an input image size of 227x227x3. Find the receptive field (with respect to the input) at the center position of layer 1, layer 2, layer 3, and layer 4, which are indicated in the diagram. For example at layer 1 the height and width is 55, the center spatial location is at 27 (round down). Your answer should provide 4 numbers to describe the receptive field Height 1, Height 2, Width 1, Width 2. Assume indexing starts at 1. You must show your work for this question.

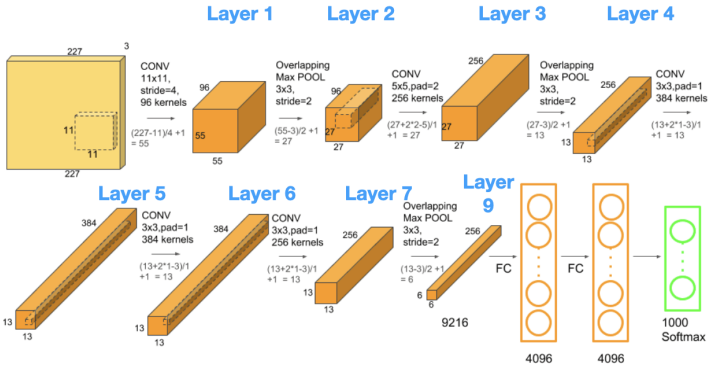

Figure 1: Labeled Alexnet

(b) **(10 points)** Download and load the pretrained alexnet network from the pytorch model repository. Write a helper function that will take a single input image and output the top prediction from alexnet using the actual class label names (e.g. "triceratops", not the label 51). This can be for example added as a label or text above the image. You can find the class labels here:
`https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a`
and some sample images here
`https://github.com/ajschumacher/imagen/tree/master/imagen`
Note you will need to resize the images to 224x224x3 to feed them into the model. Show your output for 3-5 images of your choosing.

(c) **(15 points)** We will create an adversarial example for each of 2 randomly selected images. Take 2 randomly selected images from different classes (from the link above). Now for each image select 3 classes that are not the true class and create 3 adversarial examples. Do this by optimizing the following objective: $\max_r \alpha\|r\|_1 + l(f_\theta(x+r), y_p)$, where y_p is the alternative class and f is the alexnet model and l is cross entropy. Write a function that performs this optimization. It is suggested that you use ADAM optimizer and a learning rate of 0.01. To stop the optimization you can use a criteria that checks if the sample has switched to a new class. You will need to tune a bit *alpha* and possibly other hyperparameters to get

Eugene Belilovsky: eugene.belilovsky@concordia.ca

more reasonable looking adversarial images for the samples you select. Your final image should be hard to distinguish from the original. You may adjust this optimization settings if you find ones which converge faster on the images you consider. Save your 2 adversarial examples and show that they indeed are misclassified by alexnet using your visualization tool from (b) to compare original and adversarial example and labels found by alexnet.

2. (a) (**20 points**) In this question we will train some basic RNN language model. The rest of this question can be found in `assignment2_starter_code.ipynb` which comes with the assignment.

3. (a) This question is about activation functions and vanishing/exploding gradients in recurrent neural networks (RNNs). Let $\rho : \mathcal{R} \to \mathcal{R}$ be an activation function. When the argument is a vector we apply ρ element-wise. Consider the following recurrent unit

$$\boldsymbol{h}_t = \boldsymbol{W}\rho(\boldsymbol{h}_{t-1}) + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}$$

(b) (**5 points**) Show that applying the activation function in this way is equivalent to the conventional way of applying the activation function: $\boldsymbol{g}_t = \rho(\boldsymbol{W}\boldsymbol{g}_{t-1} + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b})$ (i.e. express $\boldsymbol{g}_t$ in terms of $\boldsymbol{h}_t$). Prove it using mathematical induction. You only need to prove the induction step in this question, assuming your expression holds for time step t-1.

(c) (**15 points**) Let $\|\boldsymbol{A}\|$ denote L_2 operator norm $\boldsymbol{A}$ ($\|A\| = \max_{x:\|x\|=\|Ax\|}$. Assume ρ has bounded derivative, $|\rho'| \leq \gamma$ for some $\gamma > 0$ and for all x. We denote $\lambda_{max}(\dot{})$ the largest eigenvalue of a symmetric matrix. Show that if the largest eigenvalue of the weights is bounded by $\frac{\delta^2}{\gamma^2}$ for some $0 \leq \delta < 1$, gradients of the hidden state will vanish over time, ie

$$\lambda_{max}(\boldsymbol{W}^T\boldsymbol{W}) \leq \frac{\delta^2}{\gamma^2} \implies \left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}_0} \right\| \longrightarrow 0 \quad \texttt{as} \quad T \longrightarrow \infty$$

use the following properties of the L_2 operator norm,

$$\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\| \qquad \texttt{and} \qquad \|\boldsymbol{A}\| = \sqrt{\lambda_{max}(\boldsymbol{A}^T\boldsymbol{A})}$$

(d) (**5 points**) What could happen if $\lambda_{max}(\boldsymbol{W}^T\boldsymbol{W}) > \frac{\delta^2}{\gamma^2}$? Is this a necessary or sufficient condition for the gradient to explode? (1-2 sentences)

Eugene Belilovsky: eugene.belilovsky@concordia.ca

4. (a) Consider the self-attention operation S($\boldsymbol{X}$) defined as follows

$$\boldsymbol{A} = \frac{1}{\alpha}\boldsymbol{X}\boldsymbol{W}_q\boldsymbol{W}_k^T\boldsymbol{X}^T$$
$$S(\boldsymbol{X})_{t,:} = softmax(\boldsymbol{A}_{t,:})\boldsymbol{X}\boldsymbol{W}_v$$

Where $\boldsymbol{X} \in R^{T\times D_{in}}$, $\boldsymbol{W}_v \in R^{D_{in}\times D_v}$, $\boldsymbol{W}_k \in R^{D_{in}\times D_k}$, $\boldsymbol{W}_q \in R^{D_{in}\times D_q}$. D_{in} being the input dimensionality and T a sequence (or set) length

(b) (**5 points**) Show that for any permutation matrix $\boldsymbol{P}$, $\boldsymbol{P}S(\boldsymbol{X}) = S(\boldsymbol{P}\boldsymbol{X})$

(c) (**5 points**) What is a way you could use $S(\boldsymbol{X})$ and a linear operation to construct a permutation invariant function i.e. $\boldsymbol{W}S(\boldsymbol{P}\boldsymbol{X}) = \boldsymbol{W}S(\boldsymbol{X})$ that can provide features for e.g. a classification task and takes in an unordered set? Specifically, find a $\boldsymbol{W}$ that satisfies the condition above.

(d) (**10 points**) Consider a batched $\boldsymbol{Z} \in \mathcal{R}^{B\times T\times D}$ that is for example the output of a self-attention layer. Implement a `nn.module` or python function using functionalize that takes $\boldsymbol{Z}$ and applies a "Position wise feedforward network" layer, $H(\boldsymbol{x}) = relu(\boldsymbol{W_1}\boldsymbol{x} + \boldsymbol{b})$ which outputs a tensor of size $\mathcal{R}^{B\times T\times D}$ as commonly used in transformer models. Implement this in two ways (1) using `nn.Linear` and (2) using `nn.Conv1d` with padding 0, stride 1 and kernel size 1. Validate your implementations match with an assert statement. You may select a random Z with the size B, T, D of your choosing for validating your implementation.

Eugene Belilovsky: eugene.belilovsky@concordia.ca