Appendix A

Demonstration of Average Precision Calculation

To demonstrate the overly optimistic results from computing the AP of a raw sigmoidal output in comparison to a 0.5-thresholded prediction (binarized output), we provide the following example:

Let $y = [y_0, y_1]$ be the ground truth annotation for a truth array of batch size two (containing two samples) such that:

- $y_0 = [0, 0, 1, 1]$; and
- $y_1 = [0, 0, 0, 1]$.

A model then predicts for each sample sigmoid outputs $\hat{y}$ such that:

- $\hat{y_0} = [0.1, 0.7, 0.75, 0.8]$; and
- $\hat{y_1} = [0.3, 0.6, 0.2, 0.8]$.

The binarized output $\tilde{y}$ is then:

- $\tilde{y_0} = \hat{y_0} \geq 0.5 = [0, 1, 1, 1]$; and
- $\tilde{y_1} = \hat{y_1} \geq 0.5 = [0, 1, 0, 1]$.

Next, we can compare the calculated AP predicted for $\hat{y}$ and $\tilde{y}$, by applying Algorithm 1, shown below. The AP score we get for $\hat{y}$ is then 1.00, while the score for $\tilde{y}$ is 0.60.

Clearly, the prediction arrays in our example show high confidence (if we are to interpret the sigmoid outputs as confidence-based probabilities) for erroneous bits in comparison to the ground truth in y. Yet, the score for the sigmoid output reflects a perfect prediction, giving an overly optimistic impression of model performance. Only the binarized output score reasonably penalizes the model.

Another way of interpreting the difference is by observing how the rectangular approximation of the precision-recall (PR) curve changes when we calculate AP for $\hat{y}$ versus $\tilde{y}$. For $\hat{y}$, we obtain the following precisions and recalls for each threshold:

Algorithm 1 Average Precision Computation

This function is written explicitly for clarity and reproduces the results of scikit-learn's average precision calculation [24] with the averaging method set to "micro".

```
1:  function AVERAGEPRECISION($y_{\text{true}}, y_{\text{pred}}$)
2:      $flatPredicted \leftarrow$ FLATTEN($y_{\text{pred}}$)
3:      $uniquePredicted \leftarrow$ UNIQUE($flatPredicted$)
4:      $thresholds \leftarrow$ SORTDESCENDING($uniquePredicted$)
5:      $precisions \leftarrow list()$
6:      $recalls \leftarrow list(0)$
7:      for all $threshold \in thresholds$ do
8:          $\gamma_{\text{pred}} \leftarrow \{\text{if } (pred \geq threshold) \text{ then 1 else 0 for each } pred \text{ in } y_{\text{pred}}\}$
9:          $tp \leftarrow$ SUM($\{1 \text{ for } (b_{\text{pred}}, b_{\text{true}}) \text{ in } (\gamma_{\text{pred}}, y_{\text{true}}) \text{ if } b_{\text{pred}} = 1 \text{ and } b_{\text{true}} = 1\}$)
10:         $fp \leftarrow$ SUM($\{1 \text{ for } (b_{\text{pred}}, b_{\text{true}}) \text{ in } (\gamma_{\text{pred}}, y_{\text{true}}) \text{ if } b_{\text{pred}} = 1 \text{ and } b_{\text{true}} = 0\}$)
11:         $fn \leftarrow$ SUM($\{1 \text{ for } (b_{\text{pred}}, b_{\text{true}}) \text{ in } (\gamma_{\text{pred}}, y_{\text{true}}) \text{ if } b_{\text{pred}} = 0 \text{ and } b_{\text{true}} = 1\}$)
12:         $precision \leftarrow tp/(tp + fp)$
13:         $recall \leftarrow tp/(tp + fn)$
14:         APPEND($precisions, precision$)
15:         APPEND($recalls, recall$)
16:     end for
17:     $recallRange \leftarrow list()$
18:     for $i \leftarrow 1$ to LENGTH($recalls$) do
19:         APPEND($recallRange, recalls[i] - recalls[i-1]$)
20:     end for
21:     $ap \leftarrow$ SUM($\{p \cdot r \text{ for } (p, r) \text{ in } (precisions, recallRange)\}$)
22:     return $ap$
23: end function
```

- Precision $= [1.00, 1.00, 0.75, 0.60, 0.50, 0.43, 0.38]$; and
- Recall $= [0.00, 0.67, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00]$.

For $\tilde{y}$:

- Precision $= [0.6, 0.38]$; and
- Recall $= [0.00, 1.00, 1.00]$.

When we plot these, we get the curves displayed in Figure A.1. Since the second component in this case is a rectangle with a width of zero, this approach collapses to the micro-precision score. However, generally, the AP of a binarized output is not equivalent to its precision score.

Lastly, we may wonder if the AP for the binarized output is always strictly lesser than the AP score for its sigmoid variant. In fact, there are rare situations (at least with regard to learned models) where this presumption fails. In the scenario where the predictions are largely the mirror opposite of the ground truth, then the AP score of the binarized output may exceed that of the sigmoid output. However, as we are dealing with the evaluation of learned models and random models, it is almost always the case that the sigmoid output's AP scores are lower-bounded by the binarized output's AP scores. We demonstrate this conclusion in Figure A.2, using data generated from a sampling procedure detailed in Algorithm 2.

Algorithm 2 Noise Factor Experiment

This sampling process entails generating 100,000 synthetic annotations. The goal is to add increasing amounts of Gaussian noise to these annotations to mimic the predictions of a "learned model". As the amount added of noise approaches infinity, the predictions approach that of a random model. In this manner, we can simulate and compare the AP scores of sigmoid and binarized outputs from a spectrum ranging from the perfect model to a random model.

```
1: $nSamples \leftarrow 100000$
2: $totalIncrements \leftarrow 20$
3: $multiplier \leftarrow 7.5$
4: $noiseFactors \leftarrow list()$
5: $sigScores \leftarrow list()$
6: $binScores \leftarrow list()$
7: for $i \leftarrow 0$ to $totalIncrements$ do
8:     $noiseFactor \leftarrow i \cdot multiplier/totalIncrements$
9:     $annotations \leftarrow \textsc{RandomIntegers}(0, 1, (nSamples, 10))$
10:    $annotationsShape \leftarrow \textsc{Shape}(annotations)$
11:    $pred \leftarrow annotations + \textsc{RandomNormal}(0, noiseFactor, annotationsShape)$
12:    $sigPred \leftarrow \textsc{Clip}(pred, 0, 1)$
13:    $binPred \leftarrow sigPred > 0.5$
14:    $sigAP \leftarrow \textsc{AveragePrecision}(annotations, sigPred)$
15:    $binAP \leftarrow \textsc{AveragePrecision}(annotations, binPred)$
16:    $\textsc{Append}(noiseFactors, noiseFactor)$
17:    $\textsc{Append}(sigScores, sigAP)$
18:    $\textsc{Append}(binScores, binAP)$
19: end for
```

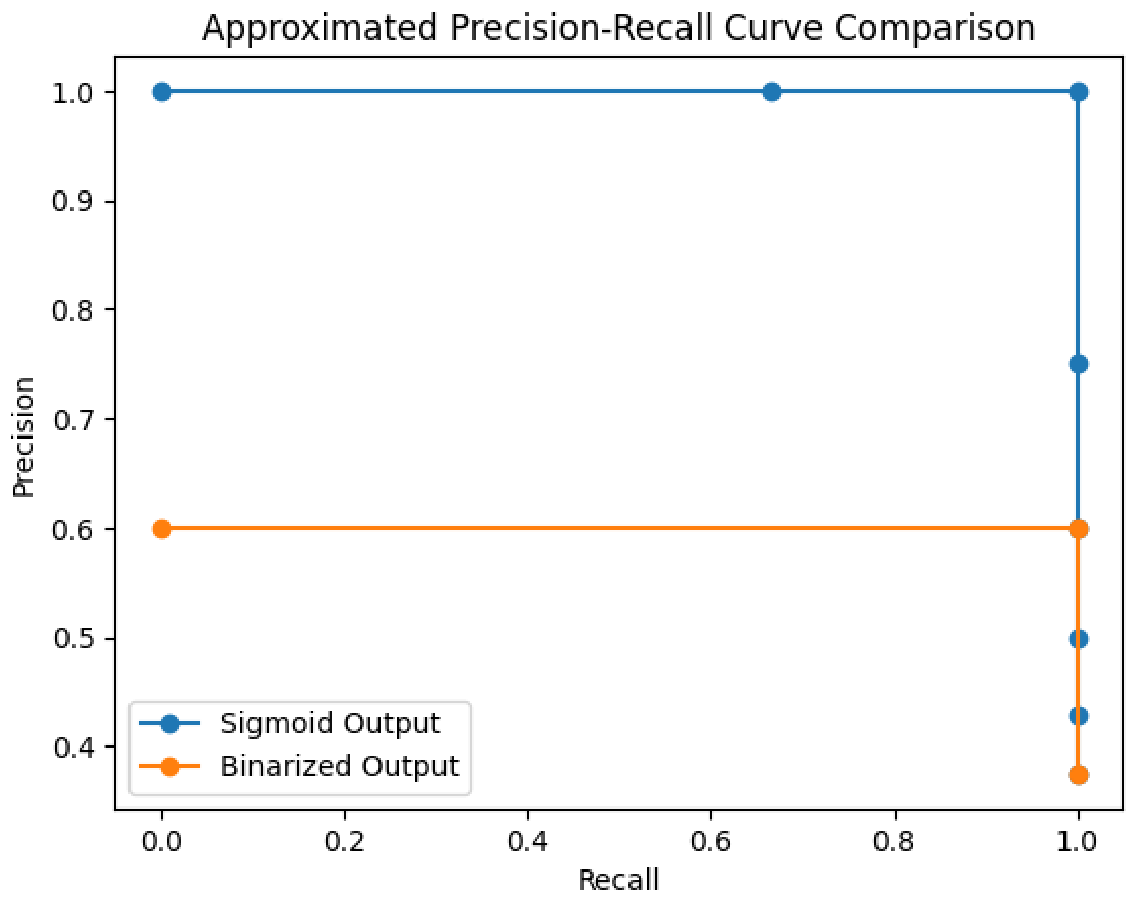

Figure A.1: **Comparison between approximated PR curves.** In our example, the precisions and recalls obtained in the intermediate steps of Algorithm 1 produce the displayed PR curves. In this particular example, both curves are comprised of only a single rectangle with non-zero width. Generally, this observation does not hold true.

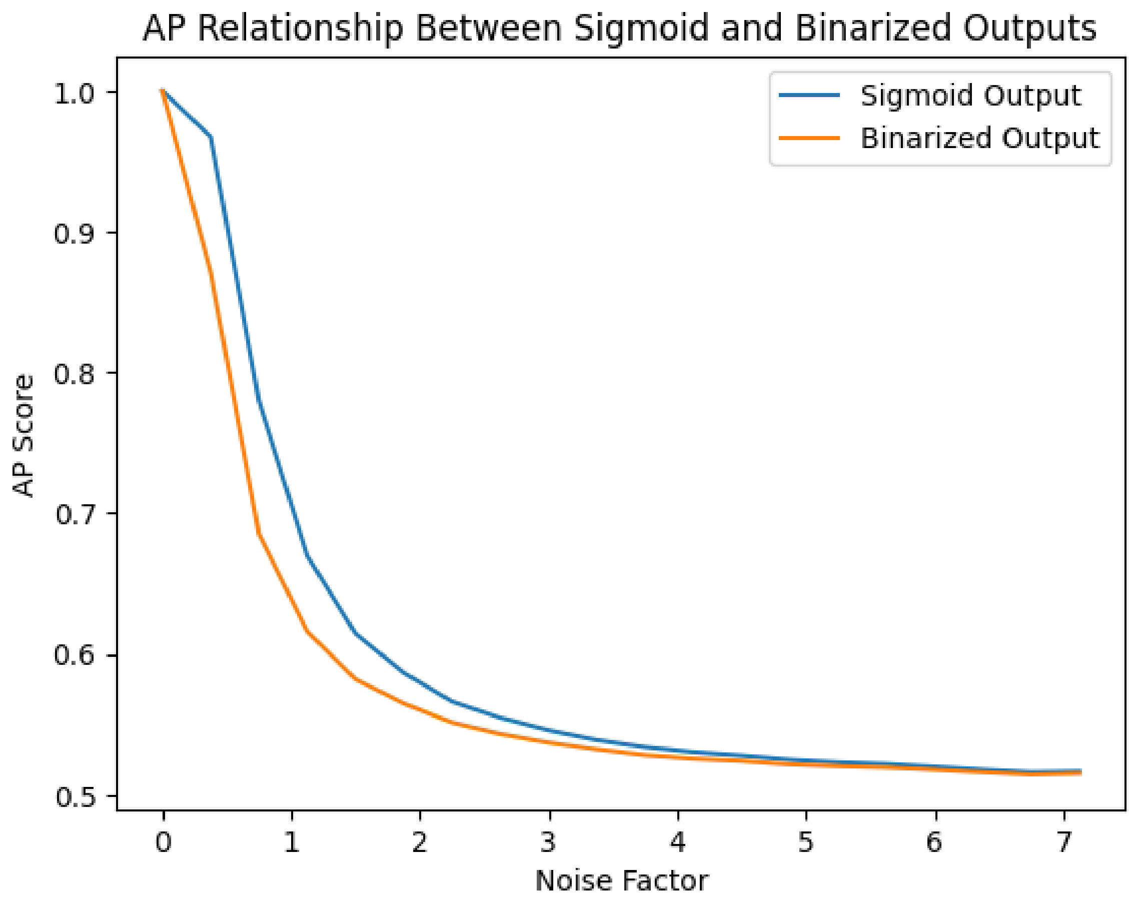

Figure A.2: **Comparison between calculated AP scores with increasing noise.** The plotted displays the output of Algorithm 2. In this experiment, as Gaussian noise is supplied to the system to mimic the descent of a perfect model into a random one, both the sigmoid and the binarized output AP scores approach 0.5, as expected. Most notably, during this entire process, the sigmoid AP scores are lower-bounded by the binarized AP scores outside of small fluctuations near convergence.