Multitask Object Detection with Dropout and Neural Code Metric Learning

Philip Pham
University of Washington
pmp10@uw.edu

Abstract

Using techniques and theory discussed in CSE 547, we try to detect objects in images with classification model and retrieve images with metric learning. The object detection task was seen as multi-label classification problem. Using multitask learning and dropout regularization, a neural network was trained to detect objects in patches. The last hidden layer of this network was used for neural code metric learning. Neighbors were retrieved with a ball tree.

1 Introduction

The COCO dataset consists of 330,000 images. Each image is annotated with bounding boxes which contain objects belonging to one of 80 categories (Lin et al., 2014). Using this dataset, I explored several techniques covered in CSE 547 including neural networks, non-convex optimization, metric learning, and nearest neighbor methods.

My objects where:

- (*object detection*) given an image, extract patches that contain an object and classify the object;
- and (*metric learning*) define a distance metric that compares patches such that patches of the same categories are close together, and patches of different categories are far apart.

In particular, I explored different neural network architecures with various learning strategies before settling on a multi-layer perceptron model with dropout. For the metric learning, I attemped to use neural codes (Babenko et al., 2014). Then, the nearest neighbor search is done using a ball tree.

2 Methods and Materials

For object detection, regions were proposed with selective search (Uijlings et al., 2013). Features for each image were given based on Ren et al. (2015). A fixed length feature vector was extracted for each patch using pooling (He et al., 2014). For an example of region proposals by selective search, see Figure 1.

Using annotations provided by the COCO API, training, validation, and test datasets were constructed from the patches from the region proposoals with intersection over union (IoU) over 0.5 with a bouding box from an annotation. That patch was then labeled with the categories from the annotated bounding box. The training set consisted of 375,453 patches from 10,000 images. The test set was 74,276 patches from 2,000 images, and the validation

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

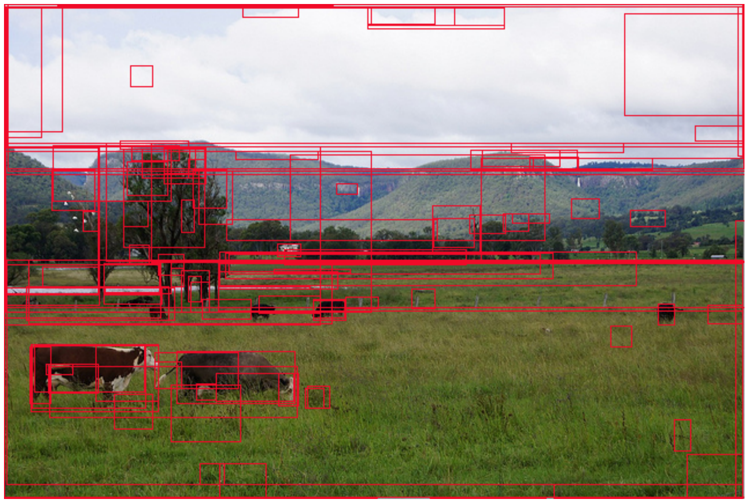

Figure 1: Top 128 region proposals by selective search for COCO image 364814.

set was 75,620 patches from 2,000 images. Each patch had 11,776 features. From here, the problem can be seen as a multi-label classification problem. Models were trained with PyTorch (Paszke et al., 2017).

The neural codes for the metric learning were extracted from the last hidden layer of the multi-layer perceptron used for object detection. To find nearest neighbors, a ball tree from `sklearn` was used (Pedregosa et al., 2011).

3 Results

The best unweighted mean average precision on the test set was 0.254112. With the learned metric, the nearest neighbor was of the same category 24% of the time.

3.1 Object Detection

The multi-layer perceptron with dropout performed best. It consisted of two hidden layers with 1,024 and 256 units, respectively. At each layer, dropout with $p = 0.5$ and ReLu activation functions were used.

3.1.1 Models

Three types of models were trained for object detection. You can see the results in Table 1. $L2$-regularization parameters and the number of hidden units were tuned to produce the smallest loss on the validation dataset. The idea behind using an MLP is that it enables weight-sharing in the hidden units. One might imagine certain animals or vehicles have common features that the model could learn together.

Model	Average Precision Score	Loss
Linear	0.1750	0.1946
2-layer MLP	0.2198	0.1824
MLP with Dropout	0.2351	0.1807

Table 1: Metrics are computed against the validation dataset.

To use multitask learning, multi-label cross entropy was used for the loss function. The $L2$-regularization parameter λ by looking at the difference between validation and training loss across various training runs. While there were large differences, regularization was increased. Eventually, validation loss stopped improving.

To obtain a further improvement, dropout regularization was applied. Dropout is a form of regularization that randomly zeroes outs a hidden unit during training (Srivastava et al., 2014). Intuitively, this prevents the network from seeing the same examples and avoids memorizing the training data.

3.1.2 Model Evaluation

3.2 Retrieval of style files

The style files for NIPS and other conference information are available on the World Wide Web at

`http://www.nips.cc/`

The file `nips_2018.pdf` contains these instructions and illustrates the various formatting requirements your NIPS paper must satisfy.

The only supported style file for NIPS 2018 is `nips_2018.sty`, rewritten for LaTeX 2_{ε}. **Previous style files for LaTeX 2.09, Microsoft Word, and RTF are no longer supported!**

The LaTeX style file contains three optional arguments: `final`, which creates a camera-ready copy, `preprint`, which creates a preprint for submission to, e.g., arXiv, and `nonatbib`, which will not load the `natbib` package for you in case of package clash.

New preprint option for 2018 If you wish to post a preprint of your work online, e.g., on arXiv, using the NIPS style, please use the `preprint` option. This will create a nonanonymized version of your work with the text "Preprint. Work in progress." in the footer. This version may be distributed as you see fit. Please **do not** use the `final` option, which should **only** be used for papers accepted to NIPS.

At submission time, please omit the `final` and `preprint` options. This will anonymize your submission and add line numbers to aid review. Please do *not* refer to these line numbers in your paper as they will be removed during generation of camera-ready copies.

The file `nips_2018.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in Sections 4, 5, and 6 below.

4 General formatting instructions

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing (leading) of 11 points. Times New Roman is the preferred typeface throughout, and will be selected for you by default. Paragraphs are separated by $1/2$ line space (5.5 points), with no indentation.

The paper title should be 17 point, initial caps/lower case, bold, centered between two horizontal rules. The top rule should be 4 points thick and the bottom rule should be 1 point thick. Allow $1/4$ inch space above and below the title to rules. All pages should start at 1 inch (6 picas) from the top of the page.

For the final version, authors' names are set in boldface, and each name is centered above the corresponding address. The lead author's name is to be listed first (left-most), and the co-authors' names (if different address) are set to follow. If there is only one co-author, list both author and co-author side by side.

Please pay special attention to the instructions in Section 6 regarding figures, tables, acknowledgments, and references.

5 Headings: first level

All headings should be lower case (except for first word and proper nouns), flush left, and bold.

First-level headings should be in 12-point type.

5.1 Headings: second level

Second-level headings should be in 10-point type.

5.1.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

6 Citations, figures, tables, references

These instructions apply to everyone.

6.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

Hasselmo, et al. (1995) investigated...

If you wish to load the `natbib` package with options, you may add the following before loading the `nips_2018` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{nips_2018}
```

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]," not "In our previous work [4]." If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form "A. Anonymous."

6.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

[1] Sample of the first footnote.

Figure 2: Sample figure caption.

Table 2: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Note that footnotes are properly typeset *after* punctuation marks.[2]

6.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

6.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 2.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 2.

7 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

[2]As in this example.

8 Preparing PDF files

Please prepare submission files with paper size "US Letter," and not, for example, "A4."

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`
- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

  ```
  \usepackage{amsfonts}
  ```

  followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

  ```
  \newcommand{\RR}{I\!\!R} %real numbers
  \newcommand{\Nat}{I\!\!N} %natural numbers
  \newcommand{\CC}{I\!\!\!\!C} %complex numbers
  ```

  Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

8.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

Babenko, A., A. Slesarev, A. Chigorin, and V. S. Lempitsky (2014). Neural codes for image retrieval. *CoRR abs/1404.1777*.

He, K., X. Zhang, S. Ren, and J. Sun (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR abs/1406.4729*.

Lin, T., M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft COCO: common objects in context. *CoRR abs/1405.0312*.

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer (2017). Automatic differentiation in pytorch. In *NIPS-W*.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Ren, S., K. He, R. B. Girshick, and J. Sun (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR abs/1506.01497*.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research 15*(1), 1929–1958.

Uijlings, J. R. R., K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders (2013, Sep). Selective search for object recognition. *International Journal of Computer Vision 104*(2), 154–171.