Multi-Task Learning on CIFAR100

Alexander Powers

Overview

- Multi-Task Learning
- Hard Parameter Sharing
- CIFAR100
- Network Architectures
- Results
- Conclusions

Multi-Task Learning

Intuition:

- Similar tasks share similar representations
- More generalizable features will be learned if the weights are shared/regularized to benefit multiple tasks

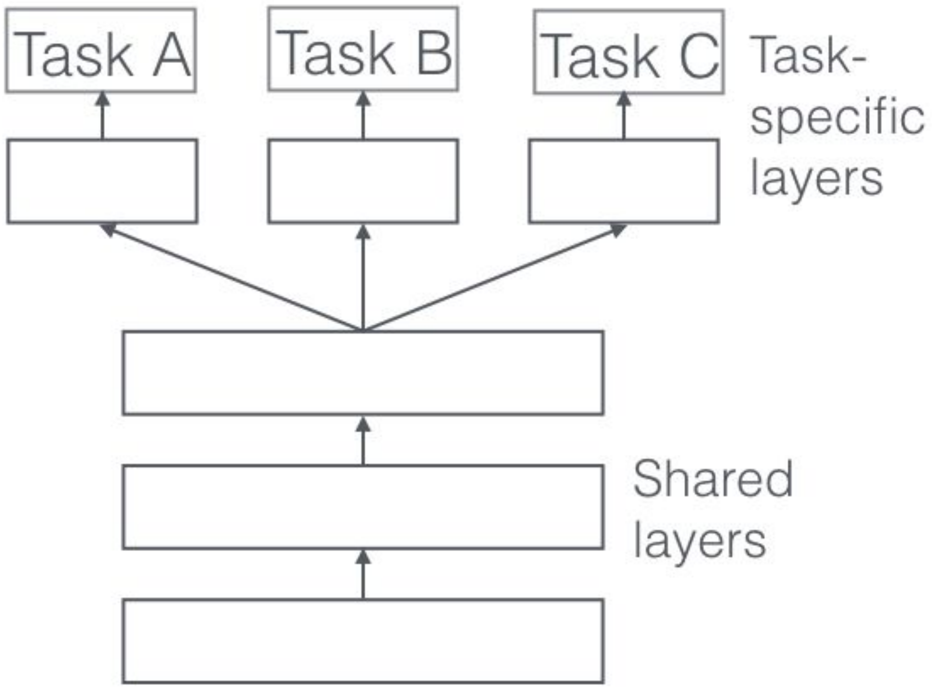
Hard Parameter Sharing

Advantages:

- Eavesdropping
- Representation Bias
- Regularization

Disadvantages:

- Task Interference
- Different Task Convergence Rates



CIFAR100

- 100 classes
- 20 superclasses each composed of 5 classes
- 500 training images and 100 test images per class
- Each image is (32 x 32 x 3) RGB

Project Goal

- Predict the coarse and fine labels simultaneously
- Develop an intuition for hard parameter sharing techniques

- THE GOAL IS NOT TO GET XY.WZ% ACCURACY

Network Architectures

Independent Networks (Control)

- Baseline network
- Tasks are learned independently

```
input_image --> conv_layers --> fc_layers --> fine_label
input_image --> conv_layers --> fc_layers --> coarse_label
```

Shared Conv Layers

- Shared convolutional layers
- Extract features that optimally represent both tasks

```
                                /--> fc_layers --> fine_label
input_image --> conv_layers
                                \--> fc_layers --> coarse_label
```

Reuse Coarse Labels

- Concatenate coarse labels to weights to predict fine labels
- Coarse labels help inform fine labels
- (knowing the coarse label does not tell us the fine label)

```
input_image   ---->   conv_layers   ---->   fc_layers_1
                                                      \
                                                     concat -> fc_layers_2  -> fine_label
                                                      /
input_image -> conv_layers -> fc_layers -> coarse_label
```

Reuse Fine Labels

- Concatenate fine labels to weights to predict coarse labels
- Fine labels help inform coarse labels
- (knowing the fine label tells us the coarse label)
- Coarse labels could be one linear layer on top of a correct fine label layer

```
input_image   ---->   conv_layers   ---->   fc_layers_1
                                                      \
                                                    concat -> fc_layers_2  -> coarse_label
                                                      /
input_image -> conv_layers -> fc_layers -> fine_label
```

Shared Conv & Coarse Reuse

- Shared convolutional layer
- Concatenation of coarse labels

```
                  / -------------> fc_layers_1
                 /                            \
input_image -> conv_layers                    concat -> fc_layers_2 -> fine_label
                 \                            /
                  \-> fc_layers -> coarse_label
```

Shared Conv & Fine Reuse

- Shared convolutional layer
- Concatenation of fine labels

```
                    / ------------> fc_layers_1
                   /                           \
input_image -> conv_layers                      concat -> fc_layers_2 -> coarse_label
                   \                           /
                    \-> fc_layers -> fine_label
```

Results

Note on Network Training

- Models used same learning rate, batch size, and number of epochs
- Models have approximately the same number of parameters

Results (Accuracy)

Architecture	Validation Fine Accuracy	Validation Coarse Accuracy
Independent	48.86%	64.16%
Shared Conv	49.09%	64.20%
Reuse Coarse	51.50%	62.29%
Reuse Fine	49.14%	68.38%
Shared Conv & Reuse Coarse	50.34%	63.91%
Shared Conv & Reuse Fine	50.91%	67.11%

Results (Loss)

Architecture	Validation Fine Loss	Validation Coarse Loss
Independent	1.96	1.15
Shared Conv	1.94	1.16
Reuse Coarse	1.83	1.21
Reuse Fine	1.95	1.07
Shared Conv & Reuse Coarse	1.86	1.16
Shared Conv & Reuse Fine	1.84	1.06

Conclusions

Conclusions

- Concatenating improves fine label classification
- Reusing coarse labels hurts coarse label accuracy (backprop issue)
- Shared convolutional weights regularize performance across task

References

[1] Richard Caruana. Multitask learning: A knowledge-based source of inductive bias. In Proceedings of the Tenth International Conference on Machine Learning, pages 41–48. Morgan Kaufmann, 1993.

[2] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017.

Thank You