Building a Convolutional Network for Brain Tumor Segmentation

Intelligent systems for pattern recognition (760AA)

Giovanni Scognamiglio

What we did

- **Research**
  - understanding convolutional architectures
  - identifying lightweight module innovations
- **Model design and implementation**
  - choosing the right architecture and modules for the task at hand
- **Test the final model**

Introduction

- **What is brain tumor segmentation?**
  - brain tumor and MRIs
  - benefits of automatically segmenting tumor area in MRIs
- **Why are neural network ideal for identifying brain tumor?**
  - fuzzy borders, hard to distinguish
  - no priors about location, shape or contrast

Introduction

- **The dataset we used**
    - created by researchers to investigate statistical relationship between tumor shape and genomic data
    - available on kaggle with 200+ notebooks with deep CNN (90% implementations are U-Net)
- **Limitations and opportunities**
    - Sporadic access to Colab GPU meant we had to take a more theoretical approach for model selection

Research - segmentation models

- **Understanding U-Net popularity**
  - what is U-Net
  - the pillars of U-Net
    - 3x3 conv
    - encoder - decoder structure
    - convolutional transpose
  - why are U-Net modules/architecture effective for medical image segmentation
  - what came after U-Net

Research - segmentation models

- **Understanding DeepLab: U-Net’s antagonist**
  - is encoding actually necessary for image segmentation?
  - what is “muti-scale contextual learning”?
  - the pillars of DeepLab:
    - dilated convolutions (a double edged sword)
    - Atrous Spatial Pyramid Pooling (ASPP)
- a note on dilated convolutions: current limitations and future possibilities

Research - lightweight models

- **Why researching lighter models?**
- **Our focus: parameter-efficient models.**
    - reduction in # of parameters and multiplications-additions (mult.adds)
- How?
    - Factorizing convolutions
    - reducing redundancy

Research - lightweight models

- **Factorizing convolutions**
    - Xception by F. Chollet
    - MobileNet family
- Reducing redundancy
    - Res-Net
    - DenseNet
    - CondenseNet

Model implementation: MobileNetv3

- Why we chose MobileNetv3
    - 1: research and papers
    - 2: code and documentation
    - 3: pre-designed segmentation head

Implementation and Results

- Model selection for two models:
  - MobileNetV3
  - MobileNetV3 with extra skip connections

2*Models	2*# of trainable parameters	Hyperparpameters					DICE coef	
		opt	lr	mom	nest	bn	DICE VAL	**DICE TEST**
MobileNetv3 + LR-ASPP	310 thousands	SGD	0.09	0.95	Yes	Yes	83.49%	**84.76%**
MobileNetv3 + LR-ASPP with extra skip connections	312 thousands	SGD	0.09	0.95	Yes	Yes	84.01%	**85.26%**
U-Net (pre-trained)	7 milion	-	-	-	-	-	71.12%	**74.12%**
U-Net (claimed)	7 milion	-	-	-	-	-	-	**82%**

Conclusion

1. Consider lightweight models before trying U-Net as default go to
2. A need for general-purpose lightweight architectures
3. MobileNetv3 proved it self as a solid lightweight general purpose architecture