

Experimental Setup, Model Selection, Overfitting, Regularization

Explaining concepts with a polynomial fitting example

Roberto Souza
Assistant Professor
Electrical and Software Engineering
Schulich School of Engineering

F2024

Outline

- Learning Goals
- Experimental setup and model selection
- Overfitting and regularization
- Metrics
- Summary

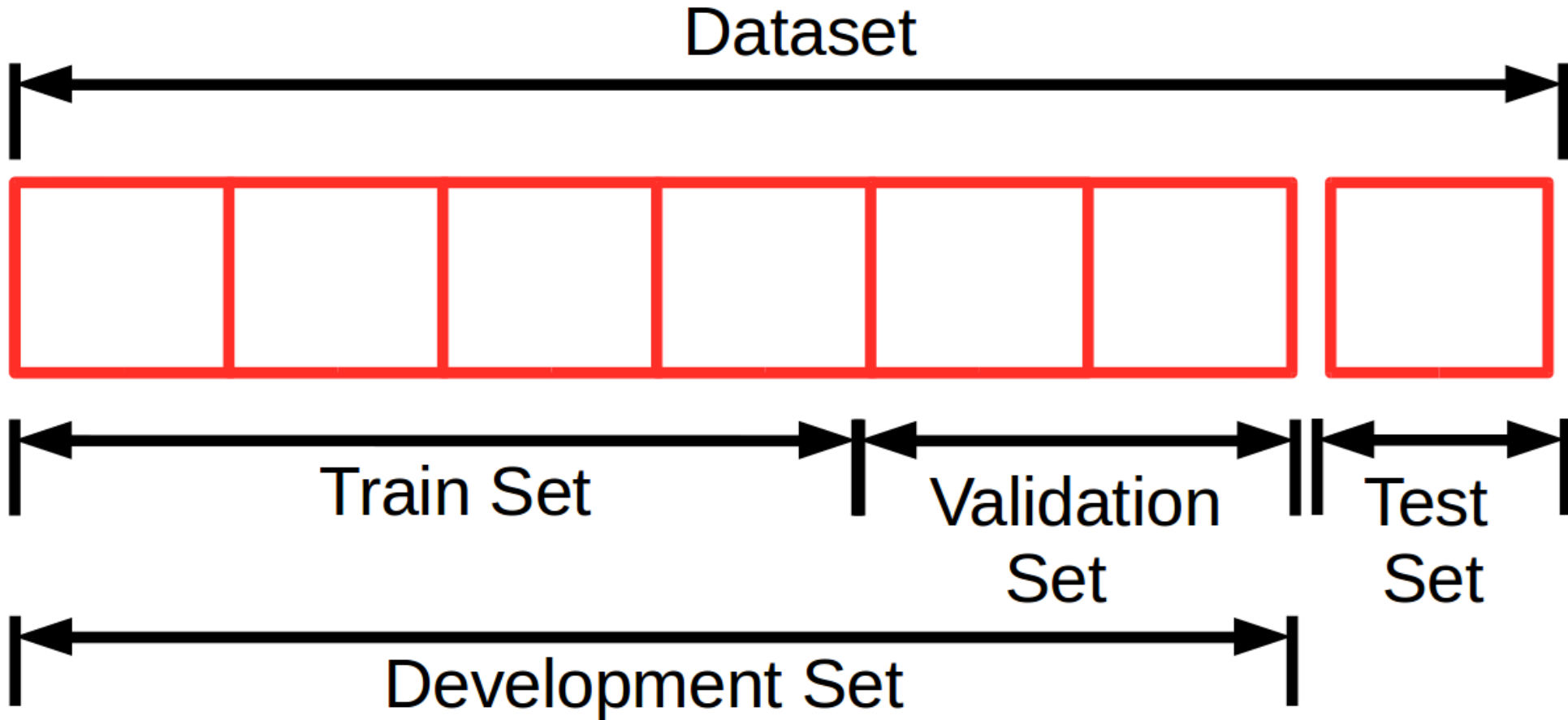
Learning Goals

- Explain how to design your experiment
- Introduce how to select your model
- Introduce the concepts of *over-fitting*, *under-fitting*, and *model generalization*.
- Introduce the concept of *regularization* for reducing model *over-fitting*.

Hands-on Tutorial

- <https://github.com/rmsouza01/deep-learning>
- **Tutorial:** [Model selection, overfitting, regularization](#)
- Based on the example presented in chapter 1 of the book: **Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA.**

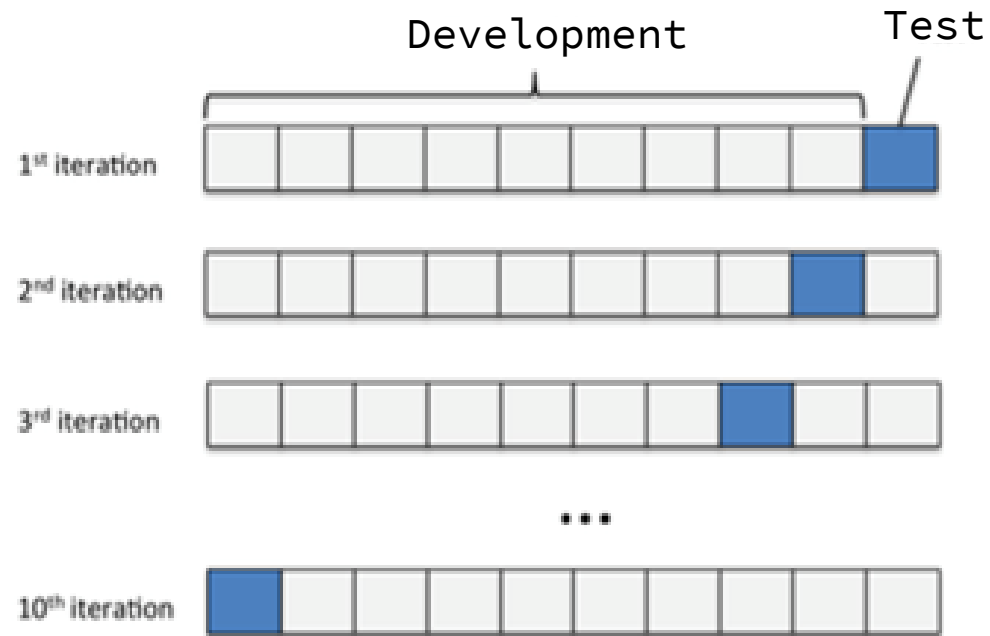
Experiment Design: Train, Validation and Test



- **Train set:** learn parameters of your models
- **Validation set:** model selection
- **Test set:** verify generalizability to unseen data

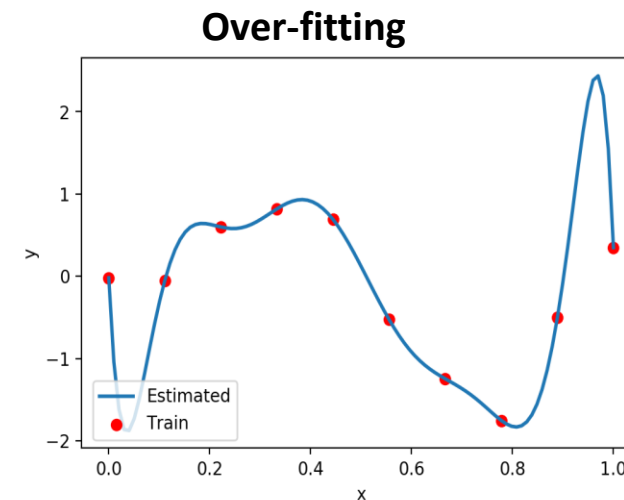
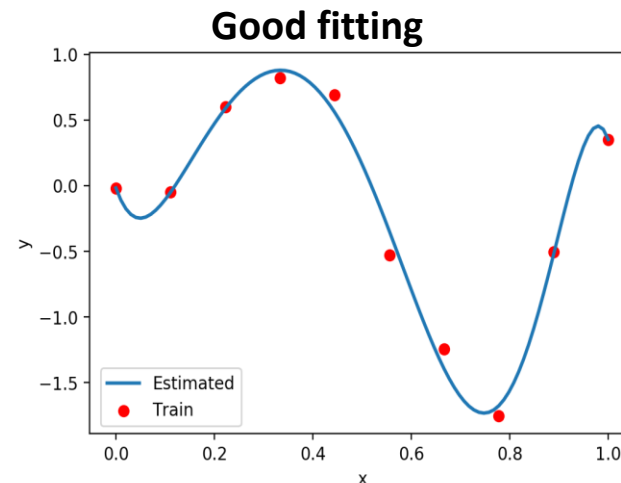
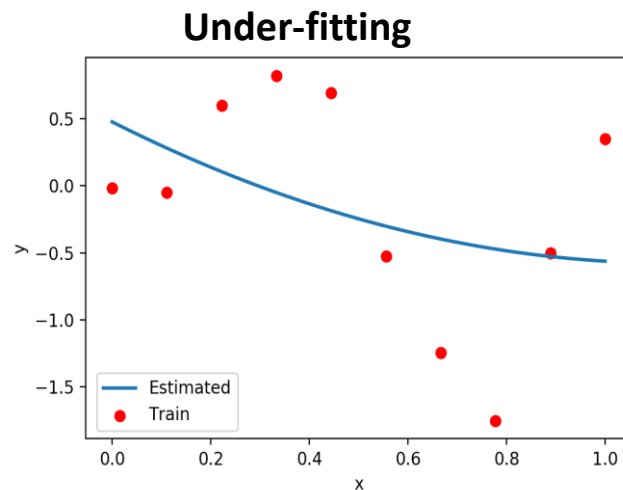
Experiment Design: k-fold cross validation

- Performs k iterations on the data
- Stratified k-fold: maintain the proportions of each class into folds (unbalance data)

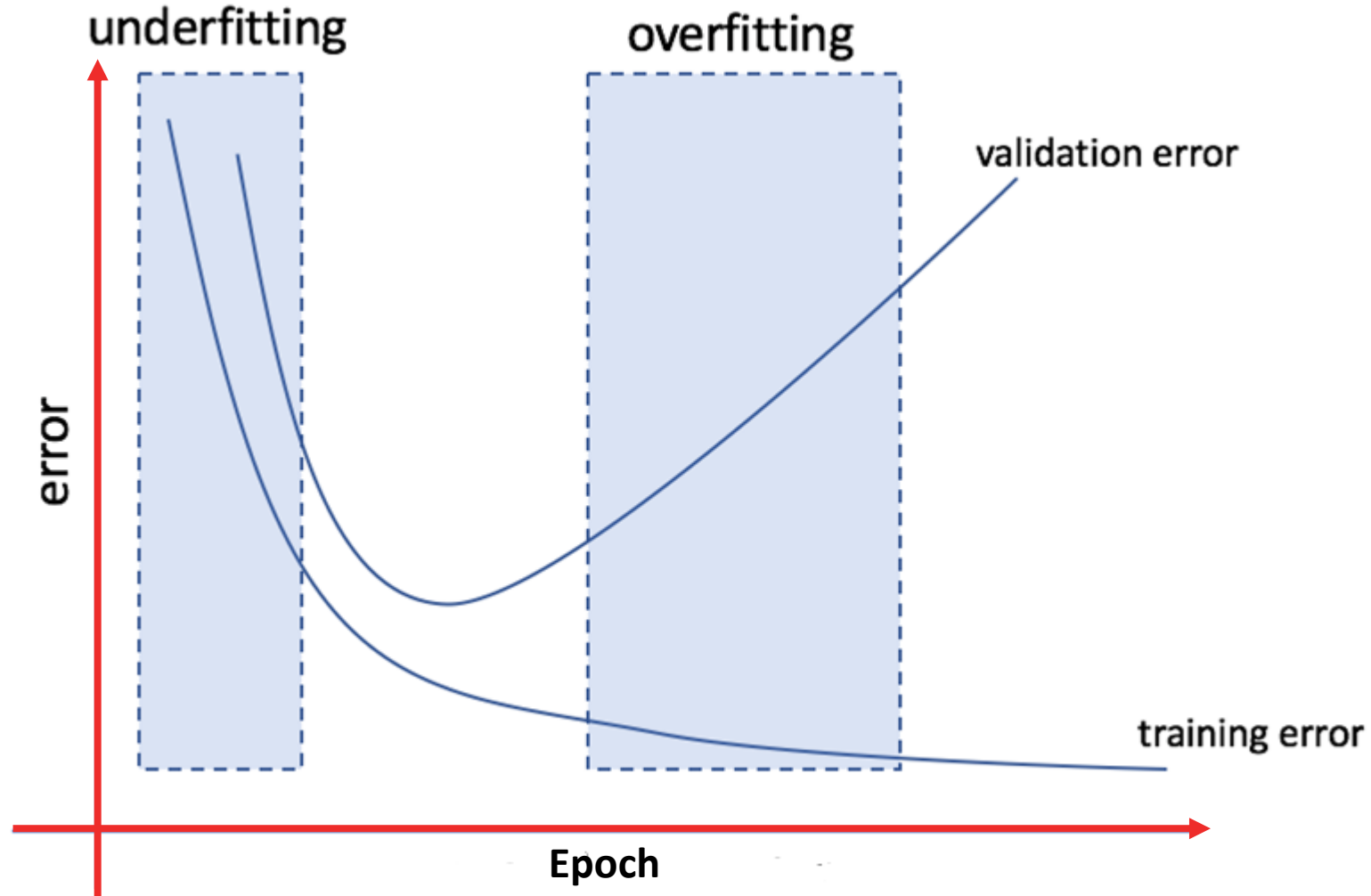


Under- and Over-fitting

- Under-fitting: too inflexible; captures no pattern
 - fitting a linear model to non-linear data
- Over-fitting: too flexible; fits to noise in the data
 - model is excessively complex ($\#features \gg \#samples$ or $\#parameters$ too high)
 - decision boundary does not generalize \rightarrow poor results for new samples



Under- and Over-fitting

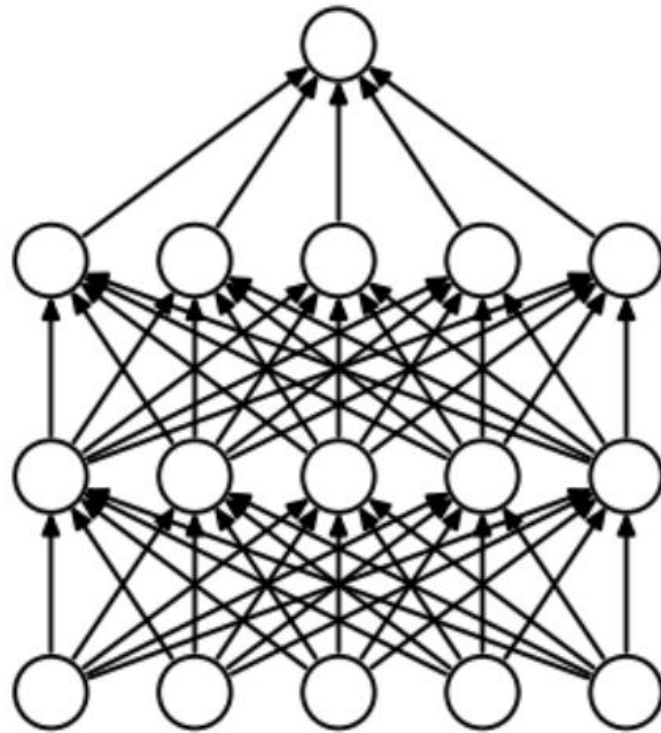


Techniques to Avoid Over-fitting

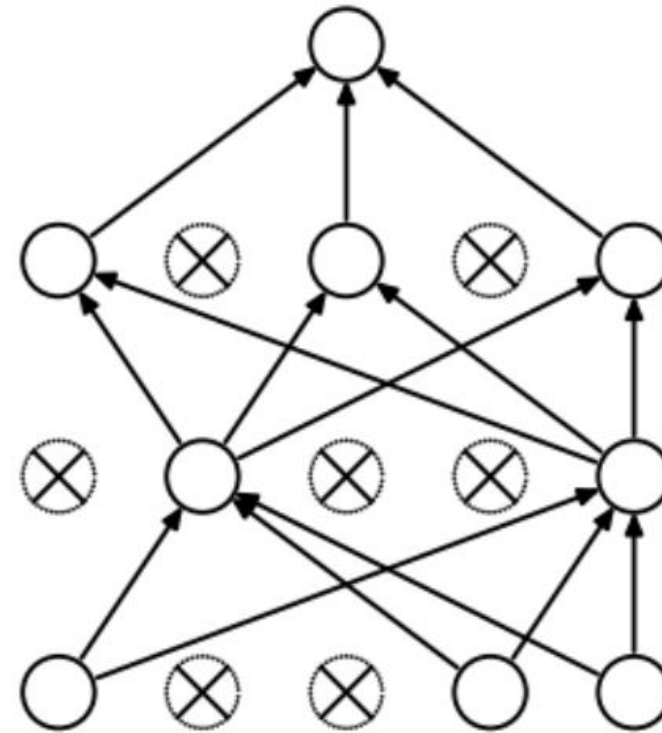
- More data
- Reduce model complexity (*i.e.*, number of trainable parameters)
- Regularization
- Dropout
- Data augmentation
- Multi-task learning

Dropout

- Learn redundant paths -> gain robustness



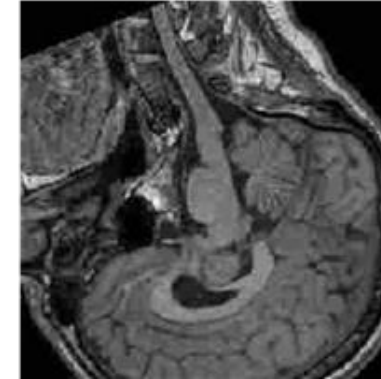
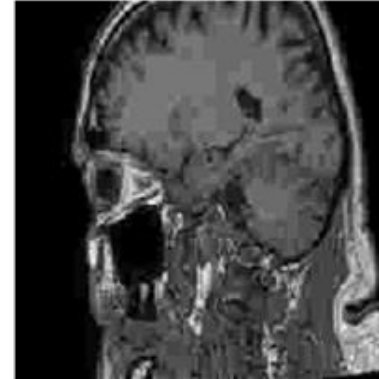
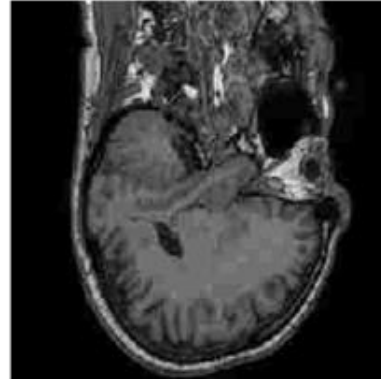
(a) Standard Neural Net



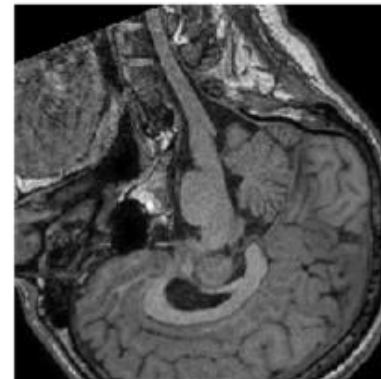
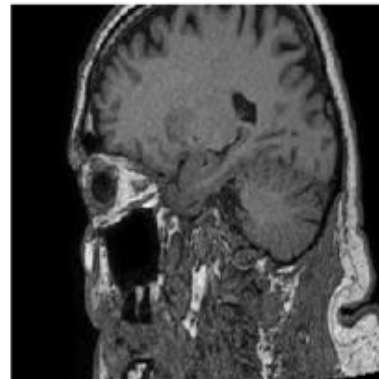
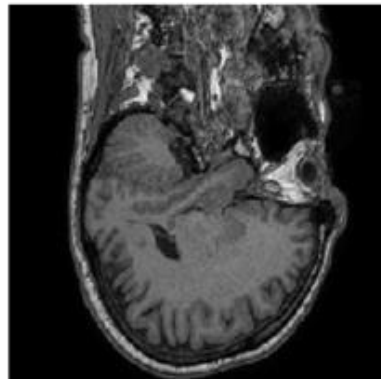
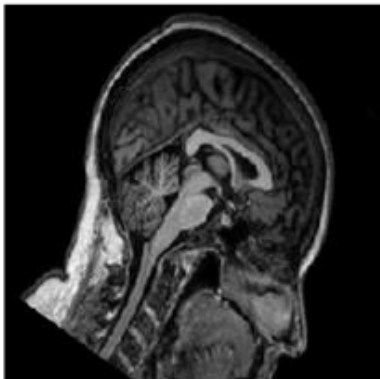
(b) After applying dropout.

Data Augmentation

- Supervised Data = Images + labels



JPEG
compressed



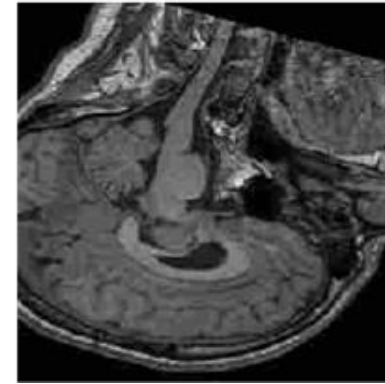
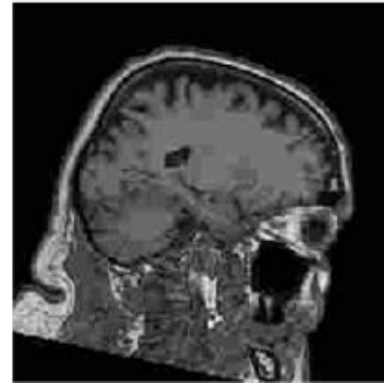
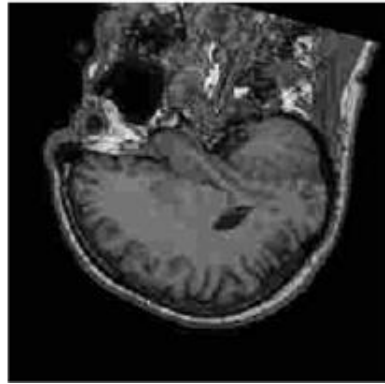
Reference

1st epoch

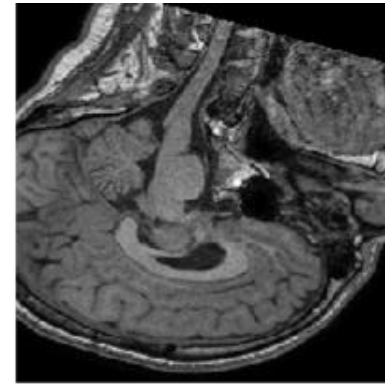
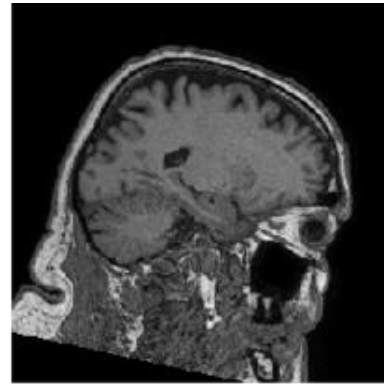
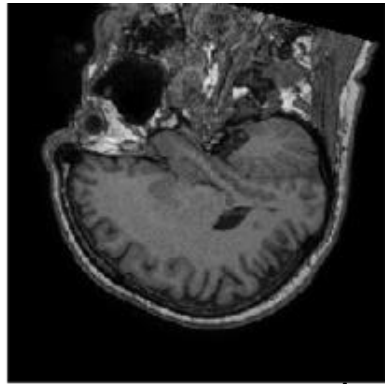
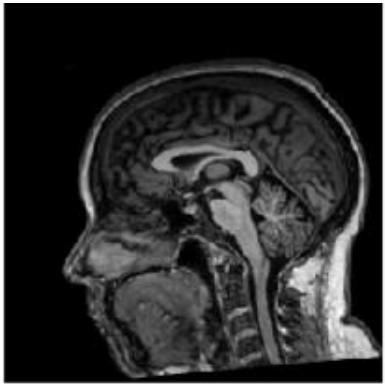
Data augmentation illustration (regression)

Data Augmentation

- Supervised Data = Images + labels



JPEG
compressed



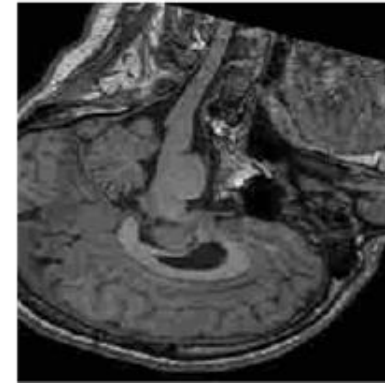
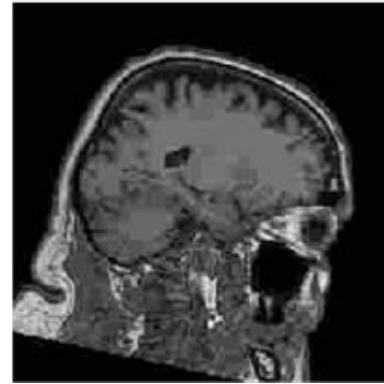
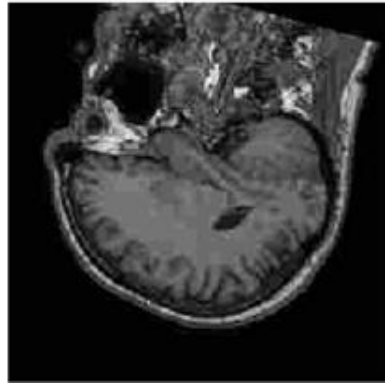
Reference

2nd epoch

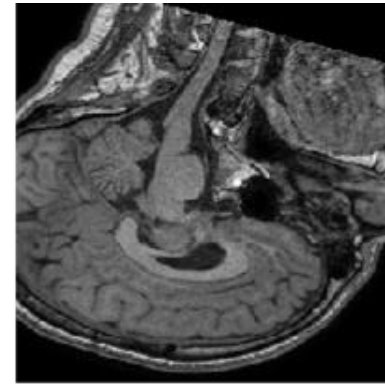
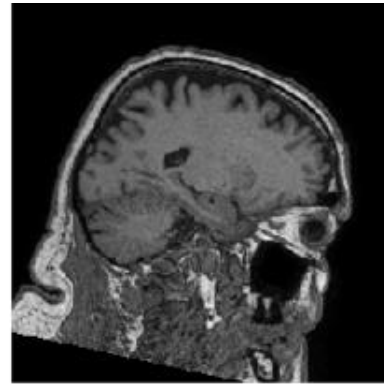
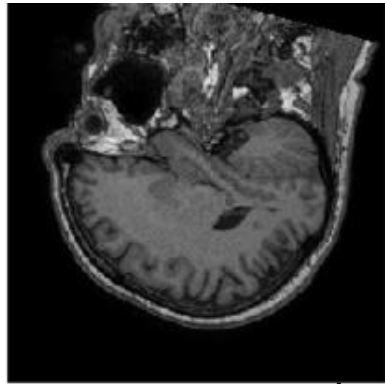
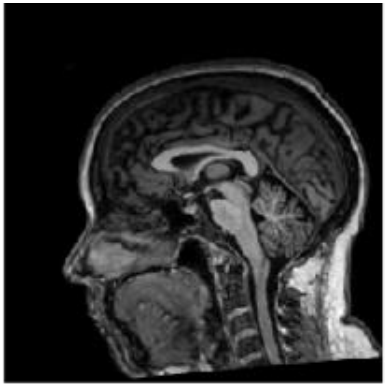
Data augmentation illustration (regression)

Data Augmentation

- Supervised Data = Images + labels



JPEG
compressed



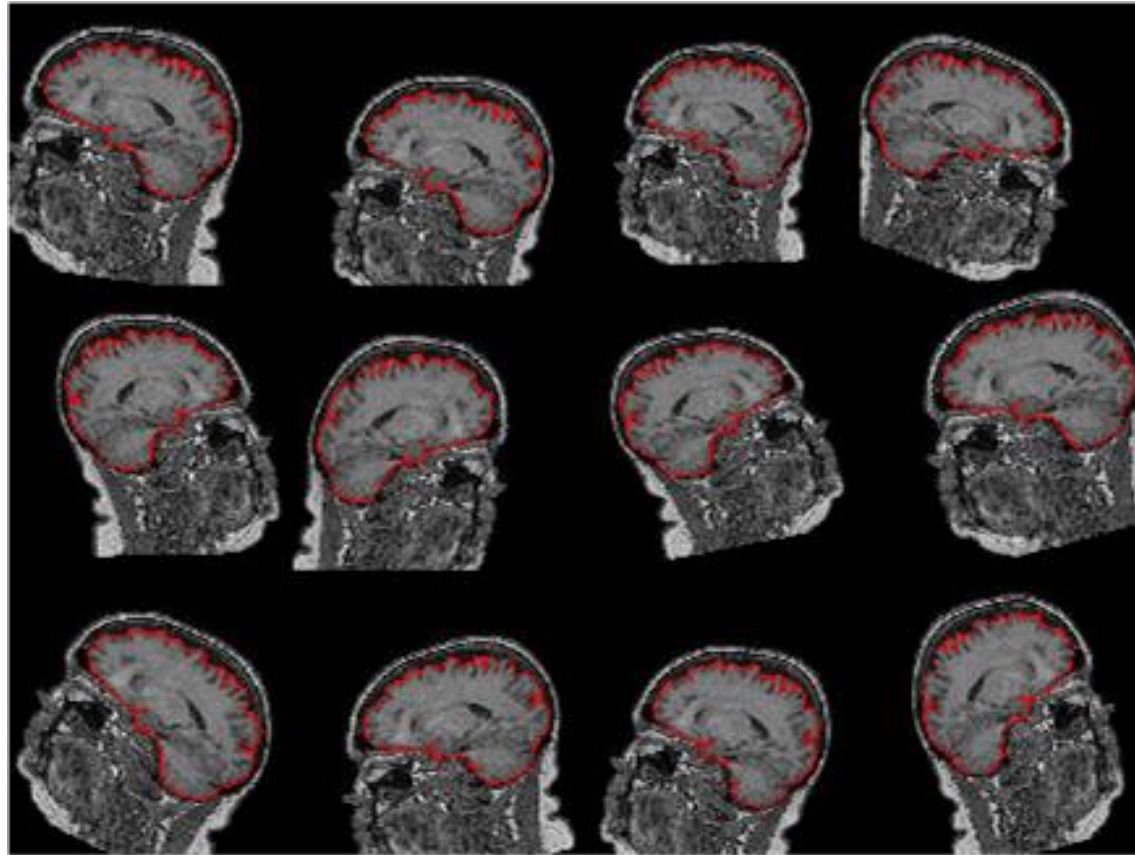
Reference

3rd epoch

Data augmentation illustration (regression)

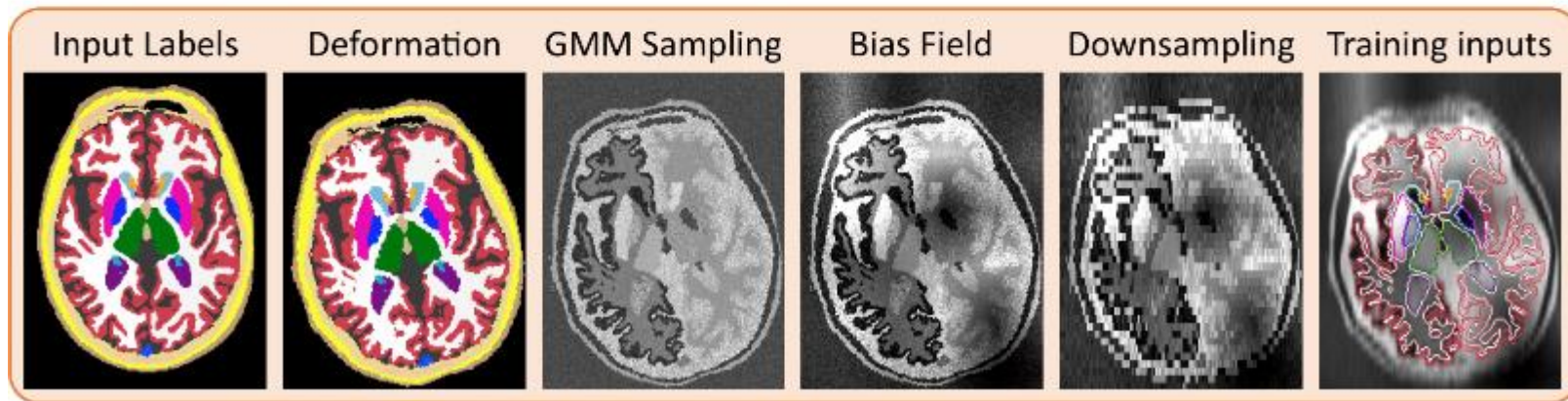
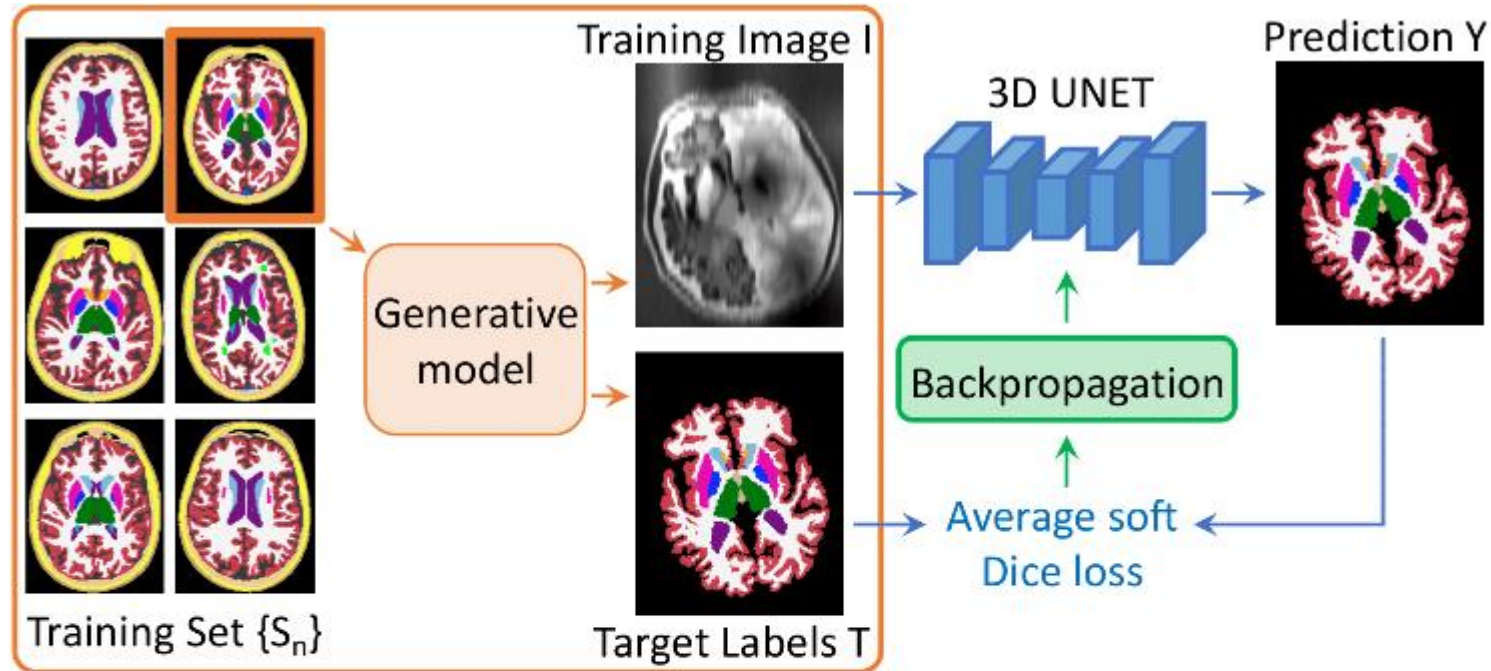
Data Augmentation

- Supervised Data = Images + labels



Data augmentation illustration (segmentation)

Domain Randomization



Metrics - Classification

- Confusion matrix

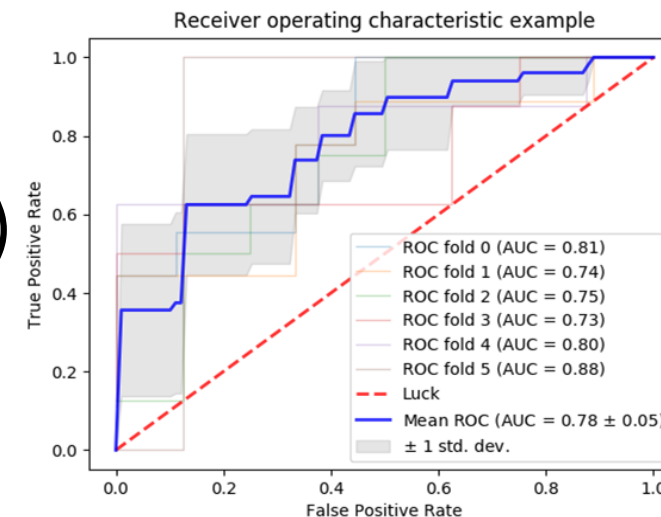
		Prediction outcome		
		positive	negative	
Actual value	positive	TP	FN	$TP + FN$
	negative	FP	TN	$FP + TN$
		$TP + FP$	$FN + TN$	

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Sensitivity = TP / P$$

$$Specificity = TN / N$$

- Receiver operating characteristic (ROC)



Metrics - Regression

- Structural Similarity (SSIM)
- Normalized Root Mean Squared Error (NRMSE)
- Peak Signal to Noise Ratio (PSNR)

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

$$\text{RMSD}(\hat{\theta}) = \sqrt{\text{MSE}(\hat{\theta})} = \sqrt{\text{E}((\hat{\theta} - \theta)^2)}.$$

$$\text{NRMSE} = \frac{\text{RMSD}}{y_{\max} - y_{\min}}$$

$$\text{MSE} = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

$$\begin{aligned} \text{PSNR} &= 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) \\ &= 20 \cdot \log_{10} \left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right) \\ &= 20 \cdot \log_{10}(\text{MAX}_I) - 10 \cdot \log_{10}(\text{MSE}) \end{aligned}$$

Summary

- For large datasets, a single train/val/test split is often sufficient
- The validation set is used for model selection
- Overfitting makes your model less generalizable to new datasets
- Model overfitting can be mitigated by employing techniques, such as regularization

Thank you!