3546 – Deep Learning

Module 4: Deep Computer Vision

News of the Week

Volunteers?

Course Syllabus

Module	Topic	Deliverables
1	Course Intro + Review	Term Project Released
2	Model Tuning	Assignment 1 Released
3	Convolutional Networks	
4	Deep Computer Vision	Assignment 1 Due, A2 Released
5	Recurrent Neural Networks	
6	Natural Language Processing	
7	Deep Models for Text	Assignment 2 Due, A3 Released
8	Representational Learning & Variational Methods	
9	Deep Generative Models	Assignment 3 Due, A4 Released
10	Speech and Music Recognition & Synthesis	
11	Term Project Presentations A	Term Project Due
12	Term Project Presentations B	Assignment 4 Due

Learning Outcomes for this Module

- Understand the applications of and modern approaches to these important computer vision tasks:
  - classification + localization
  - object detection
  - semantic segmentation
  - instance segmentation

Module 4

→ *See Jupyter Notebook for Core Content*

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 4

Supplemental Materials

Validation vs. Test Datasets

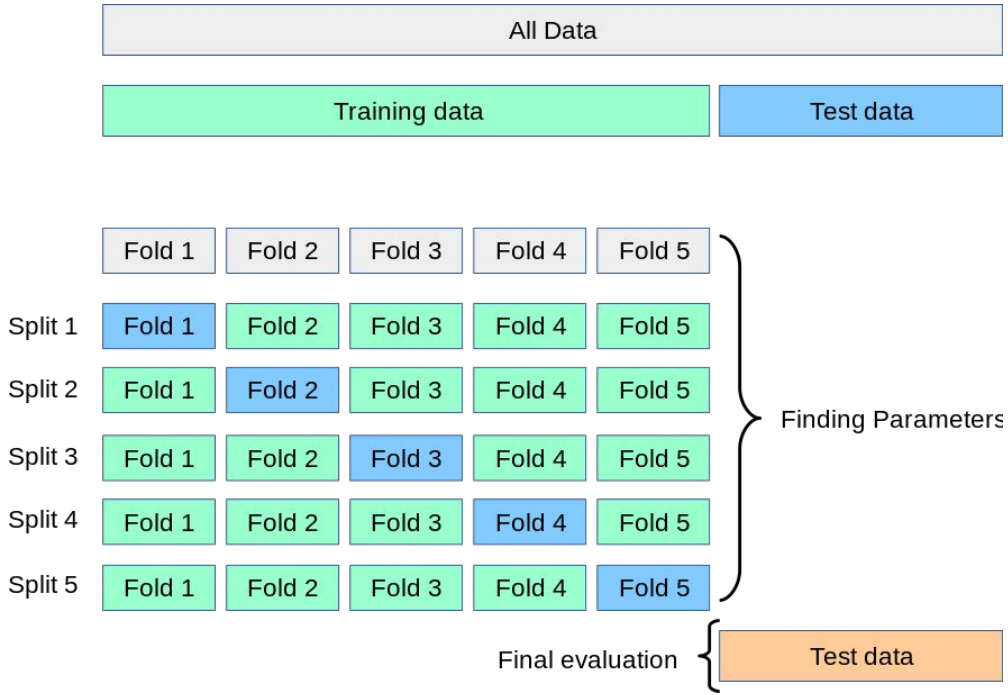
When tuning your model architecture or hyperparameters, evaluate generalization performance using a **validation** dataset, split from your training data.

Reserve the **test dataset** for final evaluation **at the very end**.

For larger networks that take a long time to train, using only a single CV split is OK.

In some examples or online, you might see folks using their test dataset as their evaluation dataset. This is not good practice. Don't get into the habit.

Example: K-Fold Cross Validation Splits



Source: https://scikit-learn.org/stable/modules/cross_validation.html

Applying Cross-Validation

Method 1:
Split manually, and pass these splits to the fit method.

```
>>> model.fit(X_train, y_train, validation_data=(X_val, y_val), …)
```

Method 2:
Ask Keras to create the split for you using the specified fraction of the total training dataset. Note that it pulls this split from the end of the training data array, so you'll want to make sure you are shuffling your data.

```
>>> model.fit(X_train, y_train, validation_split=0.20, …)
```

Applying K-Fold Cross-Validation

K-Fold CV provides a more statistically robust estimate of model performance, but is often omitted for Deep Networks that take a long time to train. You are not obligated to use this unless specifically asked.

```
>>> from sklearn.model_selection import KFold

>>> kfolds = KFold(n_splits=5, shuffle=True)

>>> for train_ids, val_ids in kfolds(X_train, y_train):

>>>       model.fit(
                X_train[train_ids],
                y_train[train_ids],
                validation_data=(
                      X_train[val_ids],
                      Y_train[val_ids]
                ),
          …)
```

Fully Convolutional Networks

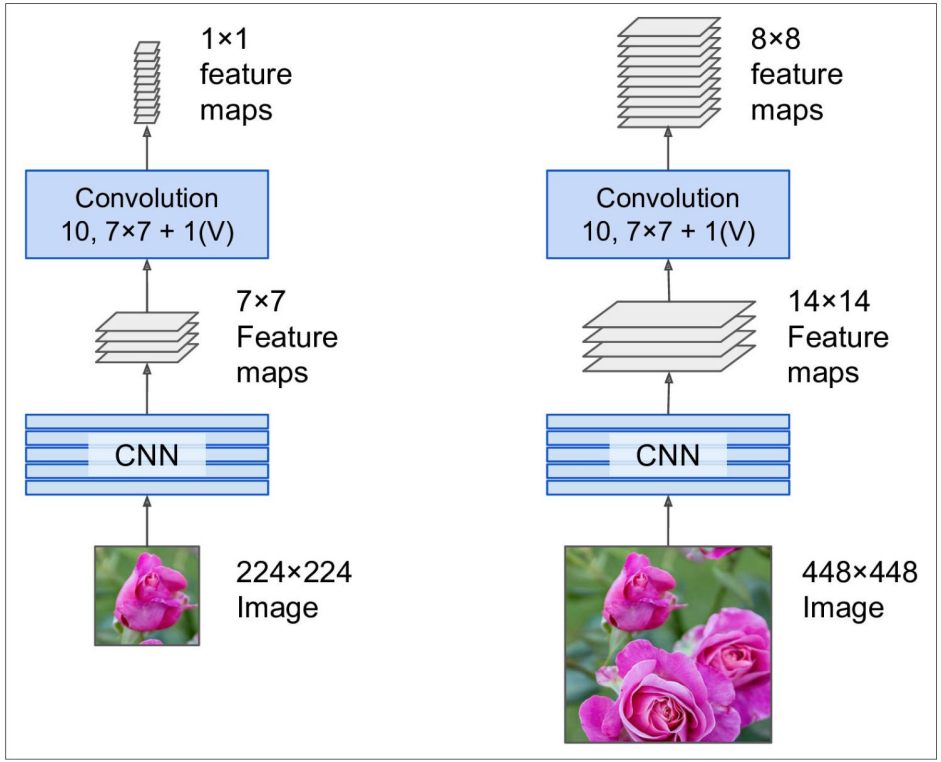


Figure 14-25. A Fully Convolutional Network Processing a Small Image (left) and a Large One (right)

Img Source: Géron, Hands-On ML, 2nd ed, Ch 14, pg 489 (2019).

Transposed Convolution

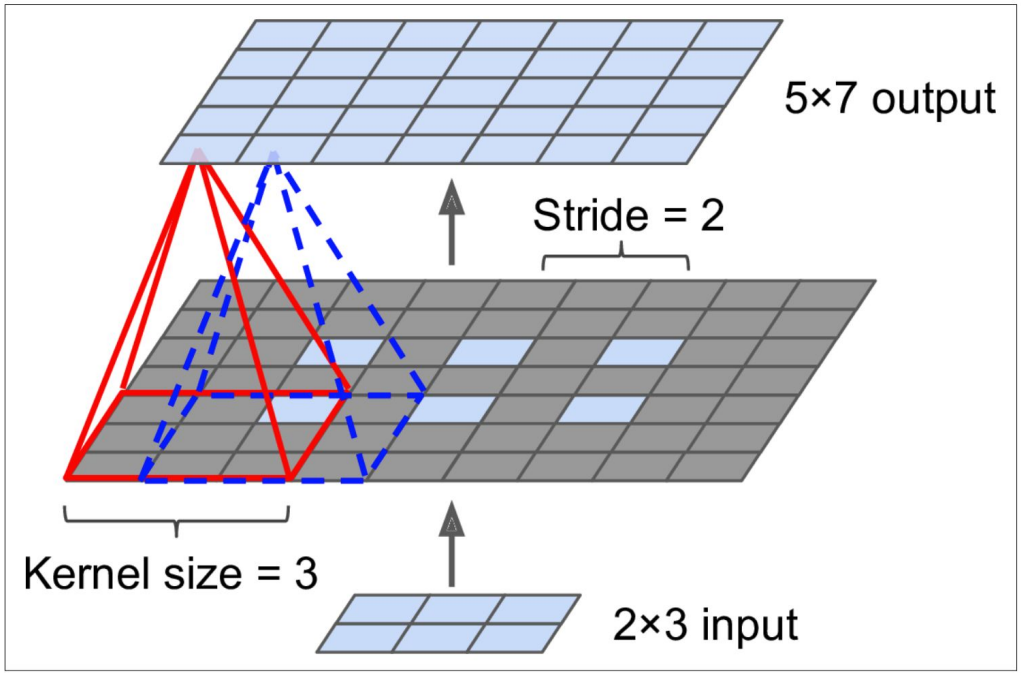
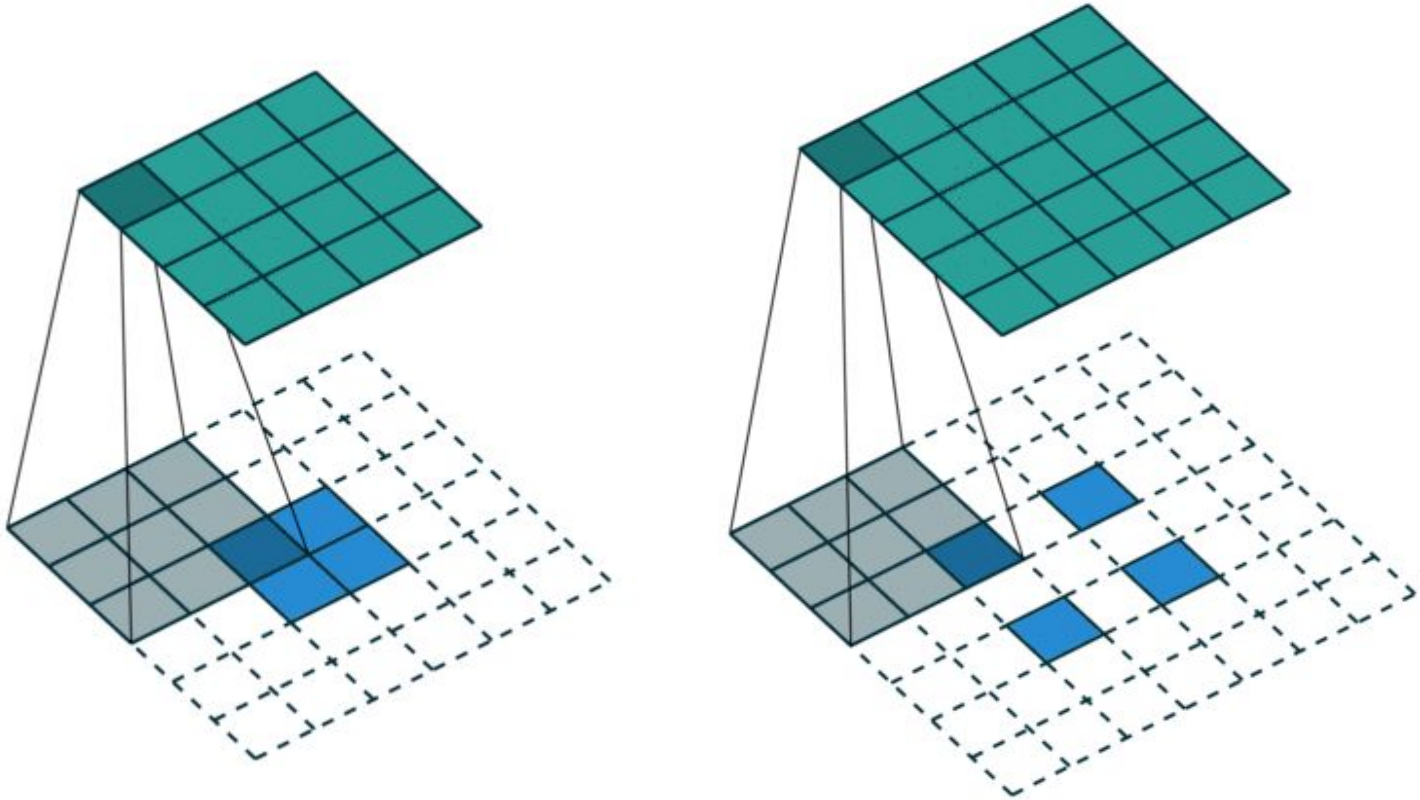


Figure 14-27. Upsampling Using a Transpose Convolutional Layer

Img Source: Géron, Hands-On ML, 2nd ed, Ch 14, pg 493 (2019).

Transposed Convolution



Module 4

Resources and Wrap-up

Homework

- Review Module 4 notebook.
- Start Assignment 2.
- Read Géron Chapter 15 (Processing Sequences Using RNNs and CNNs) to prep for next week.

Next Class

- We'll be covering Recursive Neural Networks (RNNs), which are building blocks for sequence modelling tasks, including text modelling.
- A preview of the jupyter notebook will be made available.

Any questions?

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Thank You

Thank you for choosing the University of Toronto School of Continuing Studies